



Программный комплекс Систэм Платформ

SePlatform.HMI 2.0

Справочное руководство



© ООО «СИСТЭМ СОФТ», 2022-2024. Все права защищены.

Авторские права на данный документ принадлежат ООО «СИСТЭМ СОФТ». Копирование, перепечатка и публикация любой части или всего документа не допускается без письменного разрешения правообладателя.

Содержание

1. Компоненты для работы с данными	43
1.1. Источник OPC	43
1.1.1. Свойства	43
1.1.1.1. Отображаемое имя	43
1.1.1.2. Кардинальное число	43
Значение	43
1.1.1.3. Длина	43
Примеры	44
1.1.1.4. Индекс	44
Примеры	44
1.1.1.5. Хост	45
1.1.1.6. ProgID	45
1.1.1.7. Путь	45
1.1.1.8. Родительский источник	46
1.1.1.9. Активность	46
1.1.1.10. Период обновления данных	47
1.1.1.11. Период переподписки	47
1.1.1.12. Полный тег	47
1.1.1.13. Состояние соединения	48
1.1.1.14. Тип ошибки	48
1.1.1.15. Строковое описание ошибки	49
1.1.2. События	49
1.1.2.1. OnConnectionStateChanged	49
1.1.2.2. OnErrorStateChanged	49
1.2. Элемент OPC <type>	49
1.2.1. Свойства	50
1.2.1.1. Отображаемое имя	50
1.2.1.2. Кардинальное число	50
Значение	50
1.2.1.3. Длина	50
Примеры	50
1.2.1.4. Индекс	50
Примеры	51
1.2.1.5. Источник	51
1.2.1.6. Путь	51
1.2.1.7. Значение	52
1.2.1.8. Значение в виде строки	52
1.2.1.9. Качество	53
1.2.1.10. Метка времени	53
1.2.1.11. Полный тег	54
1.2.1.12. Оформлена ли подписка?	54
1.2.1.13. Единицы измерения	55
1.2.1.14. Описание	55
1.2.1.15. Состояние подписки	56
1.2.1.16. Строковое описание состояния подписки	56
1.2.1.17. Канонический тип данных	57
1.2.2. События	57
1.2.2.1. OnVqtChanged	57
1.2.2.2. OnAdviseStateChanged	57
1.2.2.3. OnValueChanged	57
1.2.2.4. OnValueChangedAsString	58
1.2.2.5. OnQualityChanged	58
1.2.2.6. OnTimeStampChanged	58

1.2.2.7. OnUnitsChanged	58
1.2.2.8. OnDescriptionChanged	58
1.2.3. Функции	59
1.2.3.1. TrySetValue	59
1.3. Источник AP	59
1.3.1. Функции	59
1.3.1.1. Ack	59
Синтаксис	60
Примеры	60
1.3.2. Свойства	60
1.3.2.1. Отображаемое имя	60
1.3.2.2. Кардинальное число	60
Значение	60
1.3.2.3. Длина	61
Примеры	61
1.3.2.4. Индекс	61
Примеры	61
1.3.2.5. Хост	62
1.3.2.6. Порт	62
1.3.2.7. Порт истории	62
1.3.2.8. Путь	63
1.3.2.9. Источник учетных данных	63
1.3.2.10. Родительский источник	63
1.3.2.11. Таймаут операций с БД	64
1.3.2.12. Таймаут установления соединения с БД	64
1.3.2.13. Максимальный размера пула буферов	64
1.3.2.14. Общее соединение	64
1.3.2.15. Активность	65
1.3.2.16. Период переподписки	65
1.3.2.17. Имя клиента	65
1.3.2.18. Идентификатор клиента	65
1.3.2.19. Полный тег	66
1.3.2.20. Состояние соединения	66
1.3.2.21. Тип ошибки	67
1.3.2.22. Строковое описание ошибки	67
1.3.3. События	68
1.3.3.1. OnConnectionStateChanged	68
1.3.3.2. OnErrorStateChanged	68
1.4. Элемент AP <type>	68
1.4.1. Функции	68
1.4.1.1. TrySetValue	68
1.4.2. Свойства	69
1.4.2.1. Отображаемое имя	69
1.4.2.2. Кардинальное число	69
Значение	69
1.4.2.3. Длина	69
Примеры	70
1.4.2.4. Индекс	70
Примеры	70
1.4.2.5. Источник	71
1.4.2.6. Путь	71
1.4.2.7. Значение	71
1.4.2.8. Значение в виде строки	72
1.4.2.9. Качество	72
1.4.2.10. Метка времени	73
1.4.2.11. Полный тег	73

1.4.2.12. Флаг ведения истории	74
1.4.2.13. Канонический тип данных	74
1.4.2.14. Оформлена ли подписка?	74
1.4.2.15. Единицы измерения	75
1.4.2.16. Описание	75
1.4.2.17. Состояние подписки	76
1.4.2.18. Строковое описание состояния подписки	76
1.4.3. События	77
1.4.3.1. OnVqtChanged	77
1.4.3.2. OnAdviseStateChanged	77
1.4.3.3. OnValueChanged	77
1.4.3.4. OnValueChangedAsString	77
1.4.3.5. OnQualityChanged	78
1.4.3.6. OnTimeStampChanged	78
1.4.3.7. OnUnitsChanged	78
1.4.3.8. OnDescriptionChanged	78
1.4.3.9. OnHistorizingChanged	78
1.5. Учетные данные AP	79
1.5.1. Свойства	79
1.5.1.1. Отображаемое имя	79
1.5.1.2. Кардинальное число	79
Значение	79
1.5.1.3. Путь к файлу с учетными данными	79
Примеры	80
1.5.1.4. Присвоены ли учетные данные	80
Примеры	80
1.5.2. Функции	80
1.5.2.1. SetPassword	80
Синтаксис	80
Примеры	81
1.5.2.2. Save	81
Синтаксис	81
Значение	81
1.5.2.3. Load	81
Синтаксис	82
Значение	82
1.5.2.4. GetLastError	82
Синтаксис	82
1.6. Браузер источника AP	82
1.6.1. Функции	83
1.6.1.1. GetChildren	83
Синтаксис	83
Примеры	83
1.6.1.2. GetSourceNodes	83
Синтаксис	83
Параметры	84
Примеры	84
1.6.1.3. GetChildrenWithRequestId	84
Синтаксис	84
Параметры	84
Примеры	85
1.6.2. Свойства	85
1.6.2.1. Отображаемое имя	85
1.6.2.2. Кардинальное число	85
Значение	85
1.6.2.3. Источник AP	85

1.6.2.4. Вывод описания предпочтительнее имени	85
Примеры	86
1.7. Запрос алармов	86
1.7.1. Свойства	86
1.7.1.1. Отображаемое имя	86
1.7.1.2. Кардинальное число	86
Значение	86
1.7.1.3. Длина	86
Примеры	87
1.7.1.4. Индекс	87
Примеры	87
1.7.1.5. Источник	88
1.7.1.6. Режим работы	88
Значение	88
1.7.1.7. Начало исторического интервала, Конец исторического интервала	88
Пример	89
1.7.1.8. Фильтр в JSON-формате	89
Примеры	90
1.7.1.9. Фильтр исторических данных в JSON-формате (HistoricalFilter)	90
1.7.1.10. Активность	91
Значения	91
1.7.1.11. Удалять квитируемые и деактивированные события	91
Значения	91
1.7.1.12. Запрашивать список активных событий при подключении к источнику	92
Значения	92
1.7.1.13. Интервал предзагрузки истории	92
1.7.1.14. Расширяющий интервал	92
1.7.1.15. Хронология исторических событий	93
Значение	93
1.7.2. Функции	93
1.7.2.1. Reload	93
Синтаксис	93
Примеры	93
1.8. Запрос значений элемента AP	94
1.8.1. Свойства	94
1.8.1.1. Отображаемое имя	94
1.8.1.2. Кардинальное число	94
Значение	94
1.8.1.3. Элемент AP	94
1.8.1.4. Режим работы	94
1.8.1.5. Начало исторического интервала	94
1.8.1.6. Конец исторического интервала	95
1.8.1.7. Активность	95
1.8.1.8. Оперативный интервал	95
1.8.1.9. Размер оперативного буфера	95
1.8.1.10. Предзагрузка исторических данных	95
1.8.2. Функции	95
1.8.2.1. Reload	95
Синтаксис	96
Примеры	96
1.9. Запрос значений множества сигналов AP	96
1.9.1. Функции	96
1.9.1.1. Reload	96
Синтаксис	96
Примеры	96
1.9.2. Свойства	97

1.9.2.1. Отображаемое имя	97
1.9.2.2. Кардинальное число	97
Значение	97
1.9.2.3. Источник АР	97
1.9.2.4. Начало исторического интервала	97
1.9.2.5. Конец исторического интервала	97
1.9.2.6. Список сигналов	97
Примеры	98
1.10. Очередь активных алармов	99
1.10.1. Свойства	99
1.10.1.1. Отображаемое имя	99
1.10.1.2. Кардинальное число	99
Значение	99
1.10.1.3. Активность	99
Значения	99
1.10.1.4. Запрос алармов	99
1.10.1.5. Фильтр в JSON-формате	100
Примеры	101
1.10.2. Функции	101
1.10.2.1. MoveNext	101
Синтаксис	101
Возвращаемое значение	101
Примеры	102
1.10.2.2. MoveFirst	102
Синтаксис	102
Возвращаемое значение	102
Примеры	102
1.10.2.3. MoveNextOrFirst	102
Синтаксис	103
Возвращаемое значение	103
Примеры	103
1.10.2.4. CurrentAlarm	103
Синтаксис	103
Возвращаемое значение	103
Примеры	104
1.10.2.5. Reload	104
Синтаксис	104
Примеры	104
1.10.2.6. Clear	104
Синтаксис	104
Примеры	105
1.10.2.7. RemoveCurrent	105
Синтаксис	105
Примеры	105
1.10.2.8. CancelRead	105
Синтаксис	105
Примеры	106
1.10.3. События	106
1.10.3.1. OnActiveStateChanged	106
Параметры	106
Примеры	106
2. Компоненты окружения	107
2.1. Окружение: архиватор	107
2.1.1. Свойства	107
2.1.1.1. Кардинальное число	107

Значение	107
2.1.1.2. Уровень сжатия	107
2.1.1.3. Занят ли архиватор	107
2.1.2. События	108
2.1.2.1. PackFinished	108
2.1.2.2. UnpackFinished	108
2.1.3. Функции	108
2.1.3.1. BrowseAsJSON	108
2.1.3.2. Interrupt	109
2.1.3.3. AsyncPack	109
2.1.3.4. AsyncUnpack	110
2.2. Окружение: группа окон	110
2.2.1. Свойства	111
2.2.1.1. Отображаемое имя	111
2.2.1.2. Кардинальное число	111
Значение	111
2.2.1.3. Длина	111
Примеры	111
2.2.1.4. Индекс	111
Примеры	112
2.2.1.5. Количество окон в группе	112
2.2.2. События	113
2.2.2.1. WindowOpened	113
2.2.2.2. WindowClosed	113
2.2.3. Функции	113
2.2.3.1. GetWindowIDbyIndex	113
2.2.3.2. SendMessageByID	113
2.2.3.3. SendToAll	114
2.3. Окружение: окно	114
2.3.1. Свойства	114
2.3.1.1. Отображаемое имя	114
2.3.1.2. Кардинальное число	115
Значение	115
2.3.1.3. Длина	115
Примеры	115
2.3.1.4. Индекс	115
Примеры	116
2.3.1.5. Группа окон	116
2.3.2. События	117
2.3.2.1. MessageReceived	117
2.4. Окружение: сеть	117
2.4.1. Свойства	117
2.4.1.1. Отображаемое имя	117
2.4.1.2. Кардинальное число	117
Значение	117
2.4.1.3. Длина	117
Примеры	118
2.4.1.4. Индекс	118
Примеры	118
2.4.1.5. Полное имя компьютера	119
2.4.1.6. Имя компьютера	119
2.5. Окружение: файлы	120
2.5.1. Свойства	120
2.5.1.1. Отображаемое имя	120
2.5.1.2. Кардинальное число	120
Значение	120

2.5.1.3. Длина	120
Примеры	120
2.5.1.4. Индекс	121
Примеры	121
2.5.1.5. Папка текущего пользователя	121
2.5.1.6. Общая папка пользователей	122
2.5.1.7. Папка текущего проекта	122
2.5.1.8. Ошибка	123
2.5.1.9. Корневой путь	123
2.5.2. Функции	123
2.5.2.1. Exists	123
Синтаксис	123
2.5.2.2. CombinePath	124
Синтаксис	124
2.5.2.3. CreateRecursiveFolder	125
Синтаксис	125
2.5.2.4. RemoveRecursive	126
Синтаксис	126
2.5.2.5. ReadTextFile	126
Синтаксис	126
2.5.2.6. AppendTextFile	127
Синтаксис	127
2.5.2.7. ReadBinaryAsJSON	128
Синтаксис	128
2.5.2.8. WriteTextFile	129
Синтаксис	129
2.5.2.9. CanonicalPath	129
Синтаксис	129
2.5.2.10. ToGeneric	130
Синтаксис	130
2.5.2.11. ToNative	130
Синтаксис	130
2.5.2.12. PreferredSeparator	131
Синтаксис	131
2.5.2.13. ParentPath	131
Синтаксис	132
2.5.2.14. HasFileName	132
Синтаксис	132
2.5.2.15. HasParentPath	133
Синтаксис	133
2.5.2.16. IsParent	134
Синтаксис	134
2.5.2.17. AbsolutePath	135
Синтаксис	135
2.5.2.18. RelativePath	135
Синтаксис	135
2.5.2.19. FileName	136
Синтаксис	136
2.5.2.20. Extension	137
Синтаксис	137
2.5.2.21. Stem	137
Синтаксис	138
2.5.2.22. IsFile	138
Синтаксис	138
2.5.2.23. IsDirectory	139
Синтаксис	139

2.5.2.24. IsSymLink	140
Синтаксис	140
2.5.2.25. FileSize	140
Синтаксис	141
2.5.2.26. CreatedTime	141
Синтаксис	141
2.5.2.27. ModifiedTime	142
Синтаксис	142
2.6. Окружение: ошибка работы с файлами	143
2.6.1. События	143
2.6.1.1. Occured	143
2.6.2. Свойства	143
2.6.2.1. Отображаемое имя	143
2.6.2.2. Кардинальное число	143
Значение	144
2.6.2.3. Состояние	144
Значение	144
2.6.2.4. Текст	144
2.6.2.5. Код	145
2.7. Окружение: система	145
2.7.1. Свойства	145
2.7.1.1. Отображаемое имя	145
2.7.1.2. Кардинальное число	145
Значение	146
2.7.1.3. Длина	146
Примеры	146
2.7.1.4. Индекс	146
Примеры	146
2.7.1.5. Название	147
2.7.1.6. Семейство	147
2.7.1.7. Версия	148
2.7.1.8. Ядро	148
2.7.1.9. Версия ядра	148
2.7.1.10. Порядок байтов	149
Значение	149
2.8. Окружение: браузер файлов	149
2.8.1. Функции	149
2.8.1.1. Browse	149
2.8.2. Свойства	150
2.8.2.1. Отображаемое имя	150
2.8.2.2. Кардинальное число	150
Значение	150
2.8.2.3. Файлы	150
2.8.2.4. Ошибка	150
Примеры	151
2.8.2.5. Рекурсивный браунинг	151
Значение	151
2.8.2.6. Показывать в имени файлов их расширения	151
Значение	151
2.8.2.7. Отображать в браузере папки	151
Значение	152
2.8.2.8. Отображать в браузере файлы	152
Значение	152
2.8.2.9. Отображать в браузере ссылки	152
Значение	152
2.8.2.10. Показывать скрытые элементы	152

Значение	152
2.8.2.11. Показывать системные элементы	153
Значение	153
2.8.2.12. Фильтр на отображаемые объекты ФС	153
2.9. Окружение: процесс	153
2.9.1. События	153
2.9.1.1. StateChanged	153
Значение	154
2.9.1.2. Started	154
2.9.1.3. Finished	154
Значение	154
2.9.1.4. StartFailed	154
2.9.1.5. OnStdout	155
2.9.1.6. OnStderr	155
2.9.2. Функции	155
2.9.2.1. Run	155
2.9.2.2. Terminate	156
2.9.2.3. Write	156
2.9.2.4. ClearAccumulatedStdout	157
2.9.2.5. ClearAccumulatedStderr	157
2.9.3. Свойства	158
2.9.3.1. Отображаемое имя	158
2.9.3.2. Кардинальное число	158
Значение	158
2.9.3.3. Переменные среды процесса	158
2.9.3.4. ID	158
2.9.3.5. State	159
Значение	159
2.9.3.6. ExitCode	159
Значение	159
2.9.3.7. Путь к исполняемому файлу	159
2.9.3.8. Путь до файла приложения	160
2.9.3.9. Показывать окно	160
2.9.3.10. Аргументы запуска	161
2.9.3.11. Рабочий каталог	161
2.9.3.12. Кодировка	162
2.9.3.13. Накапливать стандартный вывод	162
2.9.3.14. Накопленный стандартный вывод	163
2.9.3.15. Накапливать стандартный вывод ошибок	163
2.9.3.16. Накопленный стандартный вывод ошибок	163
2.10. Окружение: этот процесс	164
2.10.1. События	164
2.10.1.1. OnRunError	164
2.10.2. Функции	164
2.10.2.1. Run	164
2.10.3. Свойства	164
2.10.3.1. Отображаемое имя	164
2.10.3.2. Кардинальное число	164
Значение	164
2.10.3.3. Переменные среды процесса	165
2.10.3.4. Рабочий каталог	165
2.10.3.5. ID	165
2.10.3.6. Путь к исполняемому файлу процесса	166
2.11. Окружение: переменные среды процесса	166
2.11.1. Свойства	166
2.11.1.1. Отображаемое имя	166

2.11.1.2. Кардинальное число	166
Значение	167
2.11.2. Функции	167
2.11.2.1. Get	167
2.11.2.2. Set	167
2.11.2.3. Append	168
2.11.2.4. Prepend	169
2.11.2.5. Remove	169
2.11.2.6. Clear	170
2.11.2.7. Reset	170
2.12. Оборудование: клавиатура	171
2.12.1. Свойства	171
2.12.1.1. Отображаемое имя	171
2.12.1.2. Кардинальное число	171
Значение	171
2.12.1.3. Длина	171
Примеры	171
2.12.1.4. Индекс	172
Примеры	172
2.12.1.5. Язык ввода	172
2.12.2. Функции	173
2.12.2.1. LocalizedLanguageName	173
2.12.2.2. KeyState	173
Буквы	174
2.13. Оборудование: монитор	176
2.13.1. Свойства	176
2.13.1.1. Отображаемое имя	176
2.13.1.2. Кардинальное число	176
Значение	177
2.13.1.3. Длина	177
Примеры	177
2.13.1.4. Индекс	177
Примеры	177
2.13.2. Функции	178
2.13.2.1. Count	178
2.13.2.2. GetResolutionX-GetResolutionY	178
2.13.2.3. GetX-GetY	179
3. Компоненты для предоставления сервисных функций	181
3.1. Узел статистики ApService	181
3.1.1. События	181
3.1.1.1. NodeStateChanged	181
Параметры	181
Примеры	182
3.1.1.2. PollingFailed	182
Параметры	182
3.1.2. Свойства	182
3.1.2.1. Отображаемое имя	182
3.1.2.2. Кардинальное число	182
Значение	182
3.1.2.3. Источник статистики ApService	183
3.1.2.4. Активность подписки	183
Значение	183
3.1.2.5. Период опроса	183
Примеры	183
3.1.2.6. Путь	183

Примеры	184
3.2. Json-браузер источника статистики ApService	184
3.2.1. События	184
3.2.1.1. GetChildrenComplete	184
Параметры	184
3.2.1.2. GetChildrenFailed	184
Параметры	184
3.2.1.3. ResolvePathComplete	185
Параметры	185
3.2.1.4. ResolvePathFailed	185
Параметры	185
3.2.2. Функции	185
3.2.2.1. GetChildren	185
Синтаксис	185
Параметры	186
Возвращаемое значение	186
Примеры	186
3.2.2.2. ResolvePath	186
Синтаксис	186
Параметры	186
Возвращаемое значение	187
Примеры	187
3.2.3. Свойства	187
3.2.3.1. Отображаемое имя	187
3.2.3.2. Кардинальное число	187
Значение	187
3.2.3.3. Источник статистики ApService	187
3.2.3.4. Включение в результат брауза узлов-папок	188
Значение	188
3.2.3.5. Включение в результат брауза узлов-секций	188
Значение	188
3.2.3.6. Включение в результат брауза узлов-значений	188
Значение	188
3.2.3.7. Сохранять в результат брауза тип значения узла	188
Значение	189
3.2.3.8. Сохранять в результат брауза значения узлов	189
Значение	189
3.2.3.9. Рекурсивный брауз	189
Значение	189
3.2.3.10. Форматирование json-строки результата брауза в читабельном виде	189
Значение	190
3.3. Rmap-браузер источника статистики ApService	190
3.3.1. Функции	190
3.3.1.1. GetChildren	190
Синтаксис	190
Параметры	190
Примеры	190
3.3.1.2. GetChildrenWithRequestId	191
Синтаксис	191
Параметры	191
Примеры	191
3.3.2. Свойства	192
3.3.2.1. Отображаемое имя	192
3.3.2.2. Кардинальное число	192
Значение	192
3.3.2.3. Источник статистики ApService	192

3.3.2.4. Брауз узлов-папок	192
Значение	192
3.3.2.5. Брауз узлов-секций	192
Значение	193
3.3.2.6. Брауз узлов-значений	193
Значение	193
3.3.2.7. Рекурсивный браузер	193
Значение	193
3.4. Источник данных статистики ApService	194
3.4.1. События	194
3.4.1.1. OnConnectionStateChanged	194
Параметры	194
Примеры	194
3.4.2. Функции	194
3.4.2.1. EscapePathPart	194
Синтаксис	195
Параметры	195
Возвращаемое значение	195
Примеры	195
3.4.2.2. GetConnectionStateAsString	195
Синтаксис	195
Параметры	196
Возвращаемое значение	196
Примеры	196
3.4.2.3. GetErrorStateAsString	196
Синтаксис	196
Параметры	196
Возвращаемое значение	196
Примеры	197
3.4.3. Свойства	197
3.4.3.1. Отображаемое имя	197
3.4.3.2. Кардинальное число	197
Значение	197
3.4.3.3. Источник учетных данных	197
3.4.3.4. Имя клиента	197
Примеры	198
3.4.3.5. Режим работы источника	198
Значение	198
3.4.3.6. Таймаут операций получения данных	198
3.4.3.7. Хост	198
3.4.3.8. Порт	199
3.4.3.9. Срез дерева статистики	199
3.4.3.10. Формат среза дерева статистики	202
Значения	202
3.4.3.11. Active	203
Значения	203
4. Запрос SQL	204
4.1. События	204
4.1.1. OnOperationFinished	204
4.1.2. OnConnectionStateChanged	204
4.2. Функции	205
4.2.1. Connect	205
4.2.1.1. Синтаксис	205
4.2.2. Execute	205
4.2.2.1. Синтаксис	205

4.2.3. Prepare	205
4.2.3.1. Синтаксис	205
4.2.4. Disconnect	206
4.2.4.1. Синтаксис	206
4.3. Свойства	206
4.3.1. Отображаемое имя	206
4.3.2. Кардинальное число	206
Значение	206
4.3.3. Строка подключения	206
Примеры	207
4.3.4. Текст запроса	207
Примеры	207
4.3.5. Состояние подключения к БД	207
Значение	207
5. Компоненты печати	209
5.1. Печать	209
5.1.1. Функции	209
5.1.1.1. Print()	209
5.1.1.2. PrintDialog()	209
5.1.1.3. PrintPreviewDialog()	210
5.1.2. Свойства	210
5.1.2.1. Отображаемое имя	210
5.1.2.2. Кардинальное число	210
Значение	210
5.1.2.3. Принтер	210
5.1.2.4. Область печати	211
Значение	211
5.1.2.5. Автоподбор ориентации	211
Значение	211
5.2. Принтер	211
5.2.1. Свойства	211
5.2.1.1. Отображаемое имя	211
5.2.1.2. Кардинальное число	212
Значение	212
5.2.1.3. Имя принтера	212
5.2.1.4. Цветовой режим	212
Значение	212
5.2.1.5. Количество копий	213
5.2.1.6. Ориентация страницы	213
Значение	213
5.2.1.7. Качество печати	213
Значение	213
5.3. Системные принтеры	214
5.3.1. События	214
5.3.1.1. LoadFinished	214
5.3.2. Функции	214
5.3.2.1. Load()	214
5.3.3. Свойства	214
5.3.3.1. Отображаемое имя	214
5.3.3.2. Кардинальное число	214
Значение	215
5.3.3.3. Имена доступных принтеров	215
5.3.3.4. Имя принтера по умолчанию	215
5.3.3.5. Информация о доступных принтерах	216
5.3.3.6. Информация о принтере по умолчанию	216

5.3.3.7. Загружена ли информация о принтерах	217
Значение	217
6. Компоненты разметки	218
6.1. Линейная разметка	218
6.1.1. Свойства	218
6.1.1.1. Отображаемое имя	218
6.1.1.2. Кардинальное число	218
Значение	218
6.1.1.3. Длина	218
Примеры	218
6.1.1.4. Индекс	219
Примеры	219
6.1.1.5. Ориентация	219
6.1.1.6. Промежуток	220
6.2. Плоская разметка	220
6.2.1. Свойства	220
6.2.1.1. Отображаемое имя	220
6.2.1.2. Кардинальное число	220
Значение	220
6.3. Вес элемента разметки	220
6.3.1. Свойства	220
6.3.1.1. Отображаемое имя	220
6.3.1.2. Кардинальное число	221
Значение	221
6.3.1.3. Значимость	221
6.4. Ограничение размеров элемента разметки	221
6.4.1. Свойства	221
6.4.1.1. Отображаемое имя	221
6.4.1.2. Кардинальное число	221
Значение	221
6.4.1.3. Минимальная ширина	221
6.4.1.4. Минимальная высота	222
6.4.1.5. Максимальная ширина	222
6.4.1.6. Максимальная высота	222
6.5. Выравнивание элемента разметки	222
6.5.1. Свойства	222
6.5.1.1. Отображаемое имя	222
6.5.1.2. Кардинальное число	222
Значение	222
6.5.1.3. Выравнивание по горизонтали	223
Значение	223
6.5.1.4. Выравнивание по вертикали	223
Значение	223
6.6. Отступы элемента разметки	224
6.6.1. Свойства	224
6.6.1.1. Отображаемое имя	224
6.6.1.2. Кардинальное число	224
Значение	224
6.6.1.3. Сверху	224
6.6.1.4. Снизу	224
6.6.1.5. Слева	225
6.6.1.6. Справа	225
7. Графические компоненты	226
7.1. Линейный градиент	226

7.1.1. Свойства	226
7.1.1.1. Отображаемое имя	226
7.1.1.2. Кардинальное число	226
Значение	226
7.1.1.3. Длина	226
Примеры	226
7.1.1.4. Индекс	227
Примеры	227
7.1.1.5. Угол поворота	227
7.1.1.6. Видимость	228
Значение	228
7.1.1.7. Непрозрачность	228
Свойства точек градиента:	229
7.1.1.8. Положение	229
7.1.1.9. Цвет	229
7.2. Радиальный градиент	230
7.2.1. Свойства	230
7.2.1.1. Отображаемое имя	230
7.2.1.2. Кардинальное число	230
Значение	230
7.2.1.3. Длина	230
Примеры	230
7.2.1.4. Индекс	230
Примеры	231
7.2.1.5. X центра и Y центра	231
7.2.1.6. Видимость	232
Значение	232
7.2.1.7. Непрозрачность	232
7.2.1.8. Размер	232
Значение	233
Свойства точек градиента:	233
7.2.1.9. Положение	233
7.2.1.10. Цвет	233
7.3. Соединительная линия	234
7.3.1. Функции	234
7.3.1.1. FormX и FormY	234
7.3.1.2. ScreenX и ScreenY	235
7.3.2. Свойства	236
7.3.2.1. Отображаемое имя	236
7.3.2.2. Кардинальное число	236
Значение	236
7.3.2.3. Координата X и Координата Y	236
7.3.2.4. Z-значение	237
7.3.2.5. Угол поворота	237
7.3.2.6. Масштаб	238
7.3.2.7. Отражение	238
7.3.2.8. Видимость	239
7.3.2.9. Включено	239
7.3.2.10. Фокус ввода	239
7.3.2.11. Всплывающая подсказка	240
7.3.2.12. Ширина и Высота	240
7.3.2.13. Декоратор	241
7.3.2.14. Цвет пера	241
7.3.2.15. Стиль пера	242
7.3.2.16. Толщина пера	242
Свойства точек	243

7.3.3. События	243
7.3.3.1. MouseMove	243
7.3.3.2. MouseClick	244
7.3.3.3. MouseRightClick	245
7.3.3.4. MouseDoubleClick	246
7.3.3.5. MousePress	246
7.3.3.6. MouseRelease	247
7.3.3.7. MouseEnter	248
7.3.3.8. MouseLeave	249
7.3.3.9. TouchMove	249
7.3.3.10. TouchPress	250
7.3.3.11. TouchRelease	251
7.3.3.12. FocusChanged	252
7.4. Место соединения	252
7.4.1. Функции	252
7.4.1.1. FormX и FormY	252
7.4.1.2. ScreenX и ScreenY	253
7.4.2. Свойства	254
7.4.2.1. Отображаемое имя	254
7.4.2.2. Кардинальное число	254
Значение	254
7.4.2.3. Координата X и Координата Y	254
7.4.2.4. Z-значение	255
7.4.2.5. Угол поворота	255
7.4.2.6. Масштаб	256
7.4.2.7. Отражение	256
7.4.2.8. Видимость	257
7.4.2.9. Включено	257
7.4.2.10. Фокус ввода	258
7.4.2.11. Всплывающая подсказка	258
7.4.2.12. Ширина и Высота	259
7.4.2.13. Декоратор	259
7.4.2.14. Тип соединения	259
7.4.2.15. Цвет пера	260
7.4.2.16. Стиль пера	260
7.4.2.17. Толщина пера	261
7.4.2.18. Цвет заливки	262
7.4.2.19. Стиль заливки	262
7.4.3. События	263
7.4.3.1. MouseMove	263
7.4.3.2. MouseClick	264
7.4.3.3. MouseRightClick	265
7.4.3.4. MouseDoubleClick	266
7.4.3.5. MousePress	267
7.4.3.6. MouseRelease	267
7.4.3.7. MouseEnter	268
7.4.3.8. MouseLeave	269
7.4.3.9. TouchMove	270
7.4.3.10. TouchPress	271
7.4.3.11. TouchRelease	271
7.4.3.12. FocusChanged	272
7.5. Линия	272
7.5.1. События	273
7.5.1.1. MouseMove	273
7.5.1.2. MouseClick	274
7.5.1.3. MouseRightClick	275

7.5.1.4. MouseDoubleClick	275
7.5.1.5. MousePress	276
7.5.1.6. MouseRelease	277
7.5.1.7. MouseEnter	278
7.5.1.8. MouseLeave	278
7.5.1.9. TouchMove	279
7.5.1.10. TouchPress	280
7.5.1.11. TouchRelease	281
7.5.1.12. FocusChanged	281
7.5.2. Свойства	282
7.5.2.1. Отображаемое имя	282
7.5.2.2. Кардинальное число	282
Значение	282
7.5.2.3. Координата X и Координата Y	282
7.5.2.4. Z-значение	283
7.5.2.5. Угол поворота	283
7.5.2.6. Масштаб	284
7.5.2.7. Отражение	284
7.5.2.8. Видимость	285
7.5.2.9. Включено	285
7.5.2.10. Фокус ввода	285
7.5.2.11. Всплывающая подсказка	286
7.5.2.12. Ширина и Высота	286
7.5.2.13. Цвет пера	287
7.5.2.14. Стиль пера	287
7.5.2.15. Толщина пера	288
7.5.2.16. Цвет заливки	289
7.5.2.17. Стиль заливки	289
Свойства точек	290
7.6. Прямоугольник	291
7.6.1. Свойства	291
7.6.1.1. Отображаемое имя	291
7.6.1.2. Кардинальное число	291
Значение	291
7.6.1.3. Длина	291
Примеры	292
7.6.1.4. Индекс	292
Примеры	292
7.6.1.5. Координата X и Координата Y	293
7.6.1.6. Координата курсора X и Координата курсора Y	294
7.6.1.7. Угол поворота	294
7.6.1.8. Масштаб	295
7.6.1.9. Видимость	295
7.6.1.10. Включено	296
7.6.1.11. Фокус ввода	296
7.6.1.12. Всплывающая подсказка	297
7.6.1.13. Ширина и Высота	297
7.6.1.14. Радиус скругления	298
7.6.1.15. Цвет пера	298
7.6.1.16. Стиль пера	299
7.6.1.17. Толщина пера	300
7.6.1.18. Цвет заливки	300
7.6.1.19. Стиль заливки	301
7.6.2. События	302
7.6.2.1. MouseMove	302
7.6.2.2. MouseClick	303

7.6.2.3. MouseRightClick	303
7.6.2.4. MouseDoubleClick	304
7.6.2.5. MousePress	305
7.6.2.6. MouseRelease	306
7.6.2.7. MouseEnter	306
7.6.2.8. MouseLeave	307
7.6.2.9. TouchMove	308
7.6.2.10. TouchPress	309
7.6.2.11. TouchRelease	310
7.6.2.12. FocusChanged	310
7.6.3. Функции	311
7.6.3.1. FormX и FormY	311
7.6.3.2. ScreenX и ScreenY	311
7.7. Эллипс	313
7.7.1. Свойства	313
7.7.1.1. Отображаемое имя	313
7.7.1.2. Кардинальное число	313
Значение	313
7.7.1.3. Длина	313
Примеры	313
7.7.1.4. Индекс	313
Примеры	314
7.7.1.5. Координата X и Координата Y	314
7.7.1.6. Координата курсора X и Координата курсора Y	315
7.7.1.7. Угол поворота	316
7.7.1.8. Масштаб	316
7.7.1.9. Видимость	317
7.7.1.10. Включено	317
7.7.1.11. Фокус ввода	318
7.7.1.12. Всплывающая подсказка	318
7.7.1.13. Ширина и Высота	319
7.7.1.14. Цвет пера	319
7.7.1.15. Стиль пера	320
7.7.1.16. Толщина пера	321
7.7.1.17. Цвет заливки	321
7.7.1.18. Стиль заливки	322
7.7.1.19. Размер сектора	323
7.7.1.20. Начало сектора	323
7.7.2. События	324
7.7.2.1. MouseMove	324
7.7.2.2. MouseClick	325
7.7.2.3. MouseRightClick	326
7.7.2.4. MouseDoubleClick	326
7.7.2.5. MousePress	327
7.7.2.6. MouseRelease	328
7.7.2.7. MouseEnter	329
7.7.2.8. MouseLeave	329
7.7.2.9. TouchMove	330
7.7.2.10. TouchPress	331
7.7.2.11. TouchRelease	332
7.7.2.12. FocusChanged	332
7.7.3. Функции	333
7.7.3.1. FormX и FormY	333
7.7.3.2. ScreenX и ScreenY	333
7.8. Изображение	334
7.8.1. Свойства	335

7.8.1.1. Отображаемое имя	335
7.8.1.2. Кардинальное число	335
Значение	335
7.8.1.3. Длина	335
Примеры	335
7.8.1.4. Индекс	335
Примеры	336
7.8.1.5. Координата X и Координата Y	336
7.8.1.6. Координата курсора X и Координата курсора Y	337
7.8.1.7. Угол поворота	338
7.8.1.8. Масштаб	338
7.8.1.9. Видимость	339
7.8.1.10. Включено	339
7.8.1.11. Фокус ввода	340
7.8.1.12. Всплывающая подсказка	340
7.8.1.13. Ширина и Высота	341
7.8.1.14. Цвет пера	341
7.8.1.15. Стил пера	342
7.8.1.16. Толщина пера	343
7.8.1.17. Файл изображения	343
7.8.1.18. Активность	343
7.8.1.19. Номер кадра	344
7.8.1.20. Скорость	344
7.8.2. События	344
7.8.2.1. MouseMove	344
7.8.2.2. MouseClick	345
7.8.2.3. MouseRightClick	346
7.8.2.4. MouseDoubleClick	347
7.8.2.5. MousePress	347
7.8.2.6. MouseRelease	348
7.8.2.7. MouseEnter	349
7.8.2.8. MouseLeave	350
7.8.2.9. TouchMove	350
7.8.2.10. TouchPress	351
7.8.2.11. TouchRelease	352
7.8.2.12. FocusChanged	353
7.8.3. Функции	353
7.8.3.1. FormX и FormY	353
7.8.3.2. ScreenX и ScreenY	354
7.9. Точка трансформации	355
7.9.1. Свойства	355
7.9.1.1. Отображаемое имя	355
7.9.1.2. Кардинальное число	355
Значение	355
7.9.1.3. Длина	355
Примеры	356
7.9.1.4. Индекс	356
Примеры	356
7.9.1.5. X и Y	357
8. Компоненты для построения пользовательского интерфейса	358
8.1. Индикатор гистограммы	358
8.1.1. Свойства	358
8.1.1.1. Отображаемое имя	358
8.1.1.2. Кардинальное число	358
Значение	358

8.1.1.3. Длина	358
Примеры	358
8.1.1.4. Индекс	359
Примеры	359
8.1.1.5. Координата X и Координата Y	359
8.1.1.6. Координата курсора X и Координата курсора Y	360
8.1.1.7. Угол поворота	361
8.1.1.8. Масштаб	361
8.1.1.9. Видимость	362
8.1.1.10. Включено	362
8.1.1.11. Фокус ввода	363
8.1.1.12. Всплывающая подсказка	363
8.1.1.13. Ширина и Высота	364
8.1.1.14. Минимальное значение	364
8.1.1.15. Максимальное значение	364
8.1.1.16. Нижнее пороговое значение	364
8.1.1.17. Верхнее пороговое значение	365
8.1.1.18. Значение	365
8.1.1.19. Ориентация	365
8.1.1.20. Цвет пера	365
8.1.1.21. Стиль пера	366
8.1.1.22. Толщина пера	367
8.1.1.23. Цвет заливки	367
8.1.1.24. Стиль заливки	368
8.1.1.25. Цвет заливки полосы	369
8.1.1.26. Стиль заливки полосы	369
8.1.1.27. Цвет заливки пониженного уровня	369
8.1.1.28. Стиль заливки пониженного уровня	369
8.1.1.29. Цвет заливки превышения уровня	369
8.1.1.30. Стиль заливки превышения уровня	370
8.1.2. События	370
8.1.2.1. MouseMove	370
8.1.2.2. MouseClick	371
8.1.2.3. MouseRightClick	371
8.1.2.4. MouseDoubleClick	372
8.1.2.5. MousePress	373
8.1.2.6. MouseRelease	374
8.1.2.7. MouseEnter	374
8.1.2.8. MouseLeave	375
8.1.2.9. TouchMove	376
8.1.2.10. TouchPress	377
8.1.2.11. TouchRelease	378
8.1.2.12. FocusChanged	378
8.1.3. Функции	379
8.1.3.1. FormX и FormY	379
8.1.3.2. ScreenX и ScreenY	379
8.2. Всплывающая подсказка	381
8.2.1. Свойства	381
8.2.1.1. Отображаемое имя	381
8.2.1.2. Кардинальное число	381
Значение	381
8.2.1.3. Длина	381
Примеры	381
8.2.1.4. Индекс	381
Примеры	382
8.2.1.5. Текст	382

8.2.1.6. Координата X и Координата Y	383
8.2.2. Функции	384
8.2.2.1. Show	384
8.2.2.2. Hide	384
8.3. Текст	384
8.3.1. Свойства	385
8.3.1.1. Отображаемое имя	385
8.3.1.2. Кардинальное число	385
Значение	385
8.3.1.3. Длина	385
Примеры	385
8.3.1.4. Индекс	385
Примеры	386
8.3.1.5. Координата X и Координата Y	386
8.3.1.6. Координата курсора X и Координата курсора Y	387
8.3.1.7. Угол поворота	388
8.3.1.8. Масштаб	388
8.3.1.9. Видимость	389
8.3.1.10. Включено	389
8.3.1.11. Фокус ввода	390
8.3.1.12. Всплывающая подсказка	390
8.3.1.13. Ширина и Высота	391
8.3.1.14. Текст	391
8.3.1.15. Шрифт	391
8.3.1.16. Цвет шрифта	392
8.3.1.17. Выравнивание текста	392
8.3.2. События	393
8.3.2.1. MouseMove	393
8.3.2.2. MouseClick	393
8.3.2.3. MouseRightClick	394
8.3.2.4. MouseDoubleClick	395
8.3.2.5. MousePress	396
8.3.2.6. MouseRelease	397
8.3.2.7. MouseEnter	397
8.3.2.8. MouseLeave	398
8.3.2.9. TouchMove	399
8.3.2.10. TouchPress	400
8.3.2.11. TouchRelease	401
8.3.2.12. FocusChanged	401
8.3.3. Функции	402
8.3.3.1. FormX и FormY	402
8.3.3.2. ScreenX и ScreenY	402
8.4. Кнопка	404
8.4.1. Свойства	404
8.4.1.1. Отображаемое имя	404
8.4.1.2. Кардинальное число	404
Значение	404
8.4.1.3. Длина	404
Примеры	404
8.4.1.4. Индекс	404
Примеры	405
8.4.1.5. Координата X и Координата Y	405
8.4.1.6. Координата курсора X и Координата курсора Y	406
8.4.1.7. Угол поворота	407
8.4.1.8. Масштаб	407
8.4.1.9. Видимость	408

8.4.1.10. Включено	408
8.4.1.11. Фокус ввода	409
8.4.1.12. Всплывающая подсказка	409
8.4.1.13. Ширина и Высота	410
8.4.1.14. Удерживаемая	410
8.4.1.15. Текст	410
8.4.1.16. Выравнивание текста	411
8.4.1.17. Шрифт	411
8.4.1.18. Цвет шрифта	411
8.4.1.19. Цвет шрифта при нажатии	412
8.4.1.20. Цвет шрифта при наведении мыши	412
8.4.1.21. Цвет заливки	413
8.4.1.22. Стилль заливки	413
8.4.1.23. Цвет заливки при нажатии	415
8.4.1.24. Стилль заливки при нажатии	416
8.4.1.25. Цвет заливки при наведении мыши	417
8.4.1.26. Стилль заливки при наведении мыши	418
8.4.1.27. Цвет пера	419
8.4.1.28. Стилль пера	420
8.4.1.29. Толщина пера	420
8.4.1.30. Цвет пера при нажатии	421
8.4.1.31. Стилль пера при нажатии	421
8.4.1.32. Толщина пера при нажатии	422
8.4.1.33. Цвет пера при наведении мыши	423
8.4.1.34. Стилль пера при наведении мыши	423
8.4.2. События	424
8.4.2.1. MouseMove	424
8.4.2.2. MouseClick	425
8.4.2.3. MouseRightClick	426
8.4.2.4. MouseDoubleClick	426
8.4.2.5. MousePress	427
8.4.2.6. MouseRelease	428
8.4.2.7. MouseEnter	429
8.4.2.8. MouseLeave	429
8.4.2.9. TouchMove	430
8.4.2.10. TouchPress	431
8.4.2.11. TouchRelease	432
8.4.2.12. FocusChanged	432
8.4.2.13. ButtonPressed	433
8.4.3. Функции	433
8.4.3.1. FormX и FormY	433
8.4.3.2. ScreenX и ScreenY	434
8.5. Курсор	435
8.5.1. Свойства	435
8.5.1.1. Отображаемое имя	435
8.5.1.2. Кардинальное число	435
Значение	435
8.5.1.3. Форма	435
8.5.1.4. Путь к файлу изображения	435
8.5.1.5. Координата X активного пикселя	436
8.5.1.6. Координата Y активного пикселя	436
8.6. Системная кнопка	437
8.6.1. Свойства	437
8.6.1.1. Отображаемое имя	437
8.6.1.2. Кардинальное число	437
Значение	437

8.6.1.3. Длина	437
Примеры	437
8.6.1.4. Индекс	437
Примеры	438
8.6.1.5. Координата X и Координата Y	438
8.6.1.6. Координата курсора X и Координата курсора Y	439
8.6.1.7. Угол поворота	440
8.6.1.8. Масштаб	440
8.6.1.9. Видимость	441
8.6.1.10. Включено	441
8.6.1.11. Фокус ввода	442
8.6.1.12. Всплывающая подсказка	442
8.6.1.13. Ширина и Высота	443
8.6.1.14. Удерживаемая	443
8.6.1.15. Текст	443
8.6.1.16. Шрифт	444
8.6.1.17. Цвет шрифта	444
8.6.2. События	444
8.6.2.1. FocusChanged	444
8.6.2.2. ButtonPressed	444
8.6.3. Функции	445
8.6.3.1. FormX и FormY	445
8.6.3.2. ScreenX и ScreenY	445
8.7. Флажок	447
8.7.1. Свойства	447
8.7.1.1. Отображаемое имя	447
8.7.1.2. Кардинальное число	447
Значение	447
8.7.1.3. Длина	447
Примеры	447
8.7.1.4. Индекс	447
Примеры	448
8.7.1.5. Координата X и Координата Y	448
8.7.1.6. Координата курсора X и Координата курсора Y	449
8.7.1.7. Угол поворота	450
8.7.1.8. Масштаб	450
8.7.1.9. Видимость	451
8.7.1.10. Включено	451
8.7.1.11. Фокус ввода	452
8.7.1.12. Всплывающая подсказка	452
8.7.1.13. Ширина и Высота	453
8.7.1.14. Текст	453
8.7.1.15. Шрифт	453
8.7.1.16. Цвет шрифта	454
8.7.1.17. Состояние	454
8.7.2. События	454
8.7.2.1. FocusChanged	454
8.7.2.2. StateChanged	455
8.7.3. Функции	455
8.7.3.1. FormX и FormY	455
8.7.3.2. ScreenX и ScreenY	456
8.8. Поле ввода	457
8.8.1. Свойства	457
8.8.1.1. Отображаемое имя	457
8.8.1.2. Кардинальное число	457
Значение	457

8.8.1.3. Длина	457
Примеры	457
8.8.1.4. Индекс	457
Примеры	458
8.8.1.5. Координата X и Координата Y	458
8.8.1.6. Координата курсора X и Координата курсора Y	459
8.8.1.7. Угол поворота	460
8.8.1.8. Масштаб	460
8.8.1.9. Видимость	461
8.8.1.10. Включено	461
8.8.1.11. Фокус ввода	462
8.8.1.12. Всплывающая подсказка	462
8.8.1.13. Ширина и Высота	463
8.8.1.14. Текст	463
8.8.1.15. Шрифт	463
8.8.1.16. Цвет шрифта	464
8.8.1.17. Выравнивание текста	464
8.8.1.18. Скрывать ввод	465
8.8.1.19. Текст заполнителя	465
8.8.2. События	465
8.8.2.1. FocusChanged	465
8.8.2.2. EditFinished	465
8.8.2.3. TextChanged	466
8.8.2.4. KeyPress	466
8.8.3. Функции	466
8.8.3.1. FormX и FormY	466
8.8.3.2. ScreenX и ScreenY	467
8.9. Индикатор прогресса	469
8.9.1. Свойства	469
8.9.1.1. Отображаемое имя	469
8.9.1.2. Кардинальное число	469
Значение	469
8.9.1.3. Длина	469
Примеры	469
8.9.1.4. Индекс	469
Примеры	470
8.9.1.5. Координата X и Координата Y	470
8.9.1.6. Координата курсора X и Координата курсора Y	471
8.9.1.7. Угол поворота	472
8.9.1.8. Масштаб	472
8.9.1.9. Видимость	473
8.9.1.10. Включено	473
8.9.1.11. Фокус ввода	474
8.9.1.12. Всплывающая подсказка	474
8.9.1.13. Ширина и Высота	475
8.9.1.14. Ориентация	475
8.9.1.15. Минимальное значение	476
8.9.1.16. Максимальное значение	476
8.9.1.17. Значение	476
8.9.1.18. Показывать текст	476
8.9.1.19. Шрифт	476
8.9.1.20. Цвет шрифта	477
8.9.2. События	477
8.9.2.1. FocusChanged	477
8.9.3. Функции	477
8.9.3.1. FormX и FormY	477

8.9.3.2. ScreenX и ScreenY	478
8.10. Ползунок	480
8.10.1. Свойства	480
8.10.1.1. Отображаемое имя	480
8.10.1.2. Кардинальное число	480
Значение	480
8.10.1.3. Длина	480
Примеры	480
8.10.1.4. Индекс	480
Примеры	481
8.10.1.5. Координата X и Координата Y	481
8.10.1.6. Координата курсора X и Координата курсора Y	482
8.10.1.7. Угол поворота	483
8.10.1.8. Масштаб	483
8.10.1.9. Видимость	484
8.10.1.10. Включено	484
8.10.1.11. Фокус ввода	485
8.10.1.12. Всплывающая подсказка	485
8.10.1.13. Ширина и Высота	486
8.10.1.14. Ориентация	486
8.10.1.15. Минимальное значение	487
8.10.1.16. Максимальное значение	487
8.10.1.17. Значение	487
8.10.2. События	487
8.10.2.1. FocusChanged	487
8.10.2.2. ValueChanged	487
8.10.3. Функции	488
8.10.3.1. FormX и FormY	488
8.10.3.2. ScreenX и ScreenY	488
8.11. Выпадающий список	490
8.11.1. Свойства	490
8.11.1.1. Отображаемое имя	490
8.11.1.2. Кардинальное число	490
Значение	490
8.11.1.3. Координата X и Координата Y	490
8.11.1.4. Z-значение	491
8.11.1.5. Координата курсора X и Координата курсора Y	491
8.11.1.6. Угол поворота	492
8.11.1.7. Масштаб	492
8.11.1.8. Отражение	493
8.11.1.9. Видимость	493
8.11.1.10. Включено	494
8.11.1.11. Фокус ввода	494
8.11.1.12. Редактируемый	495
8.11.1.13. Всплывающая подсказка	495
8.11.1.14. Ширина и Высота	495
8.11.1.15. Текст	496
8.11.1.16. Выбранный индекс	496
8.11.1.17. Количество видимых элементов	496
8.11.1.18. Режим вставки	496
8.11.1.19. Шрифт	497
8.11.1.20. Цвет шрифта	497
8.11.2. События	498
8.11.2.1. Activated	498
8.11.2.2. FocusChanged	498
8.11.2.3. SelectedChanged	498

8.11.2.4. TextChanged	498
8.11.3. Функции	498
8.11.3.1. FormX и FormY	498
8.11.3.2. ScreenX и ScreenY	499
8.11.3.3. Clear	500
8.11.3.4. GetItem	501
8.11.3.5. IndexOf	501
8.11.3.6. SetItem	502
8.11.3.7. AddItem	503
8.11.3.8. RemoveItem	503
8.12. Фрейм	504
8.12.1. Свойства	504
8.12.1.1. Отображаемое имя	504
8.12.1.2. Кардинальное число	504
Значение	504
8.12.1.3. Длина	504
Примеры	505
8.12.1.4. Индекс	505
Примеры	505
8.12.1.5. Координата курсора X и Координата курсора Y	506
8.12.1.6. Угол поворота	507
8.12.1.7. Масштаб	507
8.12.1.8. Видимость	508
8.12.1.9. Включено	508
8.12.1.10. Фокус ввода	509
8.12.1.11. Всплывающая подсказка	509
8.12.1.12. Ширина и Высота	510
8.12.1.13. Цвет пера	510
8.12.1.14. Стил пера	511
8.12.1.15. Толщина пера	512
8.12.1.16. Цвет заливки	512
8.12.1.17. Стил заливки	513
8.12.1.18. Ручное управление масштабом	514
8.12.1.19. Отображать полосы прокрутки	514
8.12.1.20. Перемещение мышью	515
8.12.2. События	515
8.12.2.1. MouseMove	515
8.12.2.2. MouseClick	516
8.12.2.3. MouseRightClick	517
8.12.2.4. MouseDoubleClick	517
8.12.2.5. MousePress	518
8.12.2.6. MouseRelease	519
8.12.2.7. MouseEnter	520
8.12.2.8. MouseLeave	520
8.12.2.9. TouchMove	521
8.12.2.10. TouchPress	522
8.12.2.11. TouchRelease	523
8.12.2.12. FocusChanged	523
8.12.3. Функции	524
8.12.3.1. FormX и FormY	524
8.12.3.2. ScreenX и ScreenY	524
8.13. Дерево	526
8.13.1. Свойства	526
8.13.1.1. Отображаемое имя	526
8.13.1.2. Кардинальное число	526
Значение	526

8.13.1.3. Длина	526
Примеры	526
8.13.1.4. Индекс	526
Примеры	527
8.13.1.5. Координата курсора X и Координата курсора Y	528
8.13.1.6. Угол поворота	528
8.13.1.7. Масштаб	529
8.13.1.8. Видимость	530
8.13.1.9. Включено	530
8.13.1.10. Фокус ввода	530
8.13.1.11. Всплывающая подсказка	531
8.13.1.12. Ширина и Высота	531
8.13.1.13. Шрифт	532
8.13.1.14. Цвет шрифта	532
8.13.1.15. Цвет фона	532
8.13.1.16. Видимость рамки	533
8.13.1.17. Число столбцов	533
8.13.2. События	533
8.13.2.1. TouchMove	533
8.13.2.2. TouchPress	534
8.13.2.3. TouchRelease	535
8.13.2.4. ItemPressed	536
8.13.2.5. FocusChanged	536
8.13.2.6. ItemClicked	536
8.13.2.7. ItemDoubleClicked	536
8.13.2.8. ItemExpanded	537
8.13.2.9. ItemCollapsed	537
8.13.3. Функции	537
8.13.3.1. FormX и FormY	537
8.13.3.2. ScreenX и ScreenY	538
8.13.3.3. AddItem	539
8.13.3.4. InsertItem	539
8.13.3.5. ChildCount	540
8.13.3.6. MoveItem	541
8.13.3.7. RemoveItem	541
8.13.3.8. RemoveChildren	542
8.13.3.9. SetColumnName	542
8.13.3.10. SetColumnWidth	543
8.13.3.11. SetColumnHidden	543
8.13.3.12. SetItemData	544
8.13.3.13. GetItemData	544
8.13.3.14. GetItemText	545
8.13.3.15. SetItemText	545
8.13.3.16. SetItemBackgroundColor	546
8.13.3.17. SetItemTextColor	546
8.13.3.18. SetItemIconPath	547
9. Компоненты для проверки ввода и преобразования форматов	548
9.1. Входной формат	548
9.1.1. Свойства	548
9.1.1.1. Отображаемое имя	548
9.1.1.2. Кардинальное число	548
Значение	548
9.1.1.3. Длина	548
Примеры	548
9.1.1.4. Индекс	549

Примеры	549
9.1.1.5. Индекс	549
Примеры	550
9.1.1.6. Ссылка на формат	550
9.1.1.7. Входное значение	550
9.1.1.8. Ошибка форматирования	551
9.1.1.9. Признак ошибки форматирования	551
9.1.1.10. Выходное значение	551
9.1.2. События	552
9.1.2.1. ValueChanged	552
9.1.2.2. FormatErrorChanged	552
9.2. Выходной формат	552
9.2.1. Свойства	552
9.2.1.1. Отображаемое имя	552
9.2.1.2. Кардинальное число	552
Значение	553
9.2.1.3. Длина	553
Примеры	553
9.2.1.4. Индекс	553
Примеры	553
9.2.1.5. Индекс	554
Примеры	554
9.2.1.6. Ссылка на формат	555
9.2.1.7. Входное значение	555
9.2.1.8. Ошибка форматирования	555
9.2.1.9. Признак ошибки форматирования	556
9.2.1.10. Выходное значение	556
9.2.2. События	557
9.2.2.1. ValueChanged	557
9.2.2.2. FormatErrorChanged	557
9.3. Проверка ввода	557
9.3.1. Свойства	557
9.3.1.1. Отображаемое имя	557
9.3.1.2. Кардинальное число	557
Значение	557
9.3.1.3. Длина	558
Примеры	558
9.3.1.4. Индекс	558
Примеры	558
9.3.1.5. Ссылка на тип проверки	559
9.3.1.6. Ввод	559
9.3.1.7. Результат проверки	559
9.3.2. События	560
9.3.2.1. ResultChanged	560
9.4. Проверка ввода регулярного выражения	560
9.4.1. Свойства	560
9.4.1.1. Отображаемое имя	560
9.4.1.2. Кардинальное число	560
Значение	560
9.4.1.3. Длина	561
Примеры	561
9.4.1.4. Индекс	561
Примеры	561
9.4.1.5. Частично верное выражение	562
9.4.1.6. Верное выражение	562
9.5. Проверка ввода целого числа	562

9.5.1. Свойства	563
9.5.1.1. Отображаемое имя	563
9.5.1.2. Кардинальное число	563
Значение	563
9.5.1.3. Длина	563
Примеры	563
9.5.1.4. Индекс	563
Примеры	564
9.5.1.5. Нижняя граница	564
9.5.1.6. Верхняя граница	564
9.6. Проверка ввода числа с плавающей запятой	564
9.6.1. Свойства	565
9.6.1.1. Отображаемое имя	565
9.6.1.2. Кардинальное число	565
Значение	565
9.6.1.3. Длина	565
Примеры	565
9.6.1.4. Индекс	565
Примеры	566
9.6.1.5. Число знаков после запятой	566
9.6.1.6. Нижняя граница	566
9.6.1.7. Верхняя граница	567
9.7. Формат булевского значения	567
9.7.1. Свойства	567
9.7.1.1. Отображаемое имя	567
9.7.1.2. Кардинальное число	567
Значение	567
9.7.1.3. Длина	567
Примеры	568
9.7.1.4. Индекс	568
Примеры	568
9.7.1.5. Значение "Истина"	569
9.7.1.6. Значение "Ложь"	569
9.8. Формат даты/времени	569
9.8.1. Свойства	569
9.8.1.1. Отображаемое имя	569
9.8.1.2. Кардинальное число	569
Значение	569
9.8.1.3. Длина	569
Примеры	570
9.8.1.4. Индекс	570
Примеры	570
9.8.1.5. Формат даты/времени	571
9.8.1.6. Локальное время	571
9.9. Формат числа с плавающей запятой	571
9.9.1. Свойства	571
9.9.1.1. Отображаемое имя	571
9.9.1.2. Кардинальное число	571
Значение	571
9.9.1.3. Длина	572
Примеры	572
9.9.1.4. Индекс	572
Примеры	572
9.9.1.5. Число знаков после запятой	573
10. Компоненты периодичного выполнения	574

10.1. Таймер	574
10.1.1. Свойства	574
10.1.1.1. Отображаемое имя	574
10.1.1.2. Кардинальное число	574
Значение	574
10.1.1.3. Длина	574
Примеры	574
10.1.1.4. Индекс	575
Примеры	575
10.1.1.5. Период	575
10.1.1.6. Активность	576
10.1.1.7. Текущее время	576
10.1.2. События	576
10.1.2.1. OnTimer	576
10.2. Мигание	577
10.2.1. Свойства	577
10.2.1.1. Отображаемое имя	577
10.2.1.2. Кардинальное число	577
Значение	577
10.2.1.3. Длина	577
Примеры	577
10.2.1.4. Индекс	578
Примеры	578
10.2.1.5. Период	578
10.2.1.6. Активность	579
10.2.1.7. Цвет пера	579
10.2.1.8. Цвет заливки	579
10.2.1.9. Цвет шрифта	579
10.2.2. События	579
10.2.2.1. Blink	579
10.3. Команда	580
10.3.1. Свойства	580
10.3.1.1. Отображаемое имя	580
10.3.1.2. Кардинальное число	580
Значение	580
10.3.1.3. Длина	580
Примеры	581
10.3.1.4. Индекс	581
Примеры	581
10.3.1.5. Включено	582
10.3.1.6. Триггер	582
10.3.2. События	582
10.3.2.1. Invoked	582
10.3.2.2. EnabledChanged	582
10.3.3. Функции	583
10.3.3.1. Invoke	583
10.4. Звук	583
10.4.1. Свойства	583
10.4.1.1. Отображаемое имя	583
10.4.1.2. Кардинальное число	583
Значение	583
10.4.1.3. Длина	584
Примеры	584
10.4.1.4. Индекс	584
Примеры	584
10.4.1.5. Число циклов	585

10.4.1.6. Громкость	585
10.4.1.7. Путь до файла	585
10.4.1.8. Активность	585
10.4.2. События	586
10.4.2.1. LoopPlayed	586
10.4.2.2. AllLoopsPlayed	586
10.5. Инициализатор	586
10.5.1. Свойства	586
10.5.1.1. Отображаемое имя	586
10.5.1.2. Кардинальное число	586
Значение	586
10.5.1.3. Длина	587
Примеры	587
10.5.1.4. Индекс	587
Примеры	587
10.5.2. События	588
10.5.2.1. Initialized	588
11. Компоненты для работы с графиками	589
11.1. График: данные	589
11.1.1. Свойства	589
11.1.1.1. Отображаемое имя	589
11.1.1.2. Кардинальное число	589
Значение	589
11.1.1.3. Длина	589
Примеры	589
11.1.1.4. Индекс	590
Примеры	590
11.1.1.5. Размер буфера данных	590
11.1.1.6. Загрузка из всех колонок данных	591
11.1.1.7. Сортировка	591
11.1.1.8. Колонка для сортировки	591
11.1.1.9. Размер данных	591
11.1.2. События	592
11.1.2.1. ReadFinished	592
11.1.2.2. SaveFinished	592
11.1.3. Функции	592
11.1.3.1. BeginReadAsync	592
11.1.3.2. CancelReadAsync	593
11.1.3.3. EnableRT	593
11.1.3.4. Clear	593
11.1.3.5. Save	593
11.1.3.6. Load	594
11.2. График: декоратор линии	594
11.2.1. Свойства	594
11.2.1.1. Отображаемое имя	594
11.2.1.2. Кардинальное число	594
Значение	594
11.2.1.3. Длина	594
Примеры	595
11.2.1.4. Индекс	595
Примеры	595
11.2.1.5. Цвет	596
11.2.1.6. Название	596
11.2.1.7. Толщина линии	596
11.2.1.8. Размер точки	596

11.2.1.9. Стиль линии	596
11.2.1.10. Активность	597
11.2.1.11. Значение минимума декоратора	597
11.2.1.12. Значение максимума декоратора	597
11.2.1.13. Номер колонки декоратора	597
11.2.1.14. Приоритет	597
11.3. График: линия	597
11.3.1. Свойства	598
11.3.1.1. Отображаемое имя	598
11.3.1.2. Кардинальное число	598
Значение	598
11.3.1.3. Длина	598
Примеры	598
11.3.1.4. Индекс	598
Примеры	599
11.3.1.5. Цвет	599
11.3.1.6. Название	599
11.3.1.7. Единица измерения	600
11.3.1.8. Толщина линии	600
11.3.1.9. Размер точки	600
11.3.1.10. Стиль линии	600
11.3.1.11. Прозрачность	600
11.3.1.12. Видимость	600
11.3.1.13. Приоритет	601
11.3.1.14. Тип графика	601
11.3.1.15. Номер колонки значений X	601
11.3.1.16. Номер колонки значений Y	601
11.3.1.17. Вывод значений пересечений с реперной линией	601
11.3.1.18. Идентификатор линии	602
11.3.2. События	602
11.3.2.1. ReadFinished	602
11.3.2.2. SaveFinished	602
11.3.3. Функции	603
11.3.3.1. SetColor	603
11.3.3.2. GetActiveControlLineLables	603
11.3.3.3. GetMinX	603
11.3.3.4. GetMaxX	603
11.3.3.5. GetMinY	604
11.3.3.6. GetMaxY	604
11.3.3.7. GetGlobalMinX	604
11.3.3.8. GetGlobalMaxX	604
11.3.3.9. GetGlobalMinY	604
11.3.3.10. GetGlobalMaxY	605
11.3.3.11. SetAxisByld	605
11.3.3.12. Save	605
11.3.3.13. Load	605
11.4. График: ось	606
11.4.1. Свойства	606
11.4.1.1. Отображаемое имя	606
11.4.1.2. Кардинальное число	606
Значение	606
11.4.1.3. Длина	606
Примеры	607
11.4.1.4. Индекс	607
Примеры	607
11.4.1.5. Цвет	608

11.4.1.6. Ориентация оси	608
11.4.1.7. Выравнивание	608
11.4.1.8. Тип данных	608
11.4.1.9. Количество делений	609
11.4.1.10. Тип масштабирования	609
11.4.1.11. Логарифмическая шкала	609
11.4.1.12. Единицы измерения	609
11.4.1.13. Положение единиц измерения	610
11.4.1.14. Формат меток	610
11.4.1.15. Формат метки времени	610
11.4.1.16. Минимум	610
11.4.1.17. Максимум	610
11.4.1.18. Название оси	611
11.4.1.19. Показывать единицы измерения	611
11.4.1.20. Положение названия оси	611
11.4.1.21. Видимость оси	611
11.4.1.22. Шрифт	612
11.4.1.23. Коэффициент масштабирования	612
11.4.2. События	612
11.4.2.1. OnRangeChange	612
11.4.2.2. OnClickValue	612
11.4.2.3. OnWheelMouse	612
11.4.3. Функции	613
11.4.3.1. setAxisType	613
11.4.3.2. SetInterval	613
11.4.3.3. GetID	613
11.5. График: поле графика	614
11.5.1. Свойства	614
11.5.1.1. Отображаемое имя	614
11.5.1.2. Кардинальное число	614
Значение	614
11.5.1.3. Длина	614
Примеры	614
11.5.1.4. Индекс	614
Примеры	615
11.5.1.5. Угол поворота	615
11.5.1.6. Масштаб	616
11.5.1.7. Видимость	616
11.5.1.8. Включено	617
11.5.1.9. Всплывающая подсказка	617
11.5.1.10. Ширина и Высота	618
11.5.1.11. Размер точки курсора	618
11.5.1.12. Толщина линии курсора	618
11.5.1.13. Режим курсора	618
11.5.1.14. Цвет фона	619
11.5.1.15. Цвет курсора	619
11.5.1.16. Цвет активной реперной линии	619
11.5.1.17. Толщина активной реперной линии	619
11.5.1.18. Стилль линии курсора	619
11.5.1.19. Таймлайн	619
11.5.1.20. Легенда	620
11.5.1.21. Выравнивание легенды	620
11.5.1.22. Экстраполяция	620
11.5.1.23. Перемещать выноски репера	620
11.5.1.24. Шрифт	621
11.5.1.25. Коэффициент масштабирования по X, %	621

11.5.1.26. Коэффициент масштабирования по Y, %	621
11.5.1.27. Управление выравниванием	621
11.5.1.28. Автообновление	621
11.5.2. События	622
11.5.2.1. OnWheelMouse	622
11.5.3. Функции	622
11.5.3.1. Update	622
11.5.3.2. ResetZoom	622
11.6. График: реперная линия	623
11.6.1. Свойства	623
11.6.1.1. Отображаемое имя	623
11.6.1.2. Кардинальное число	623
Значение	623
11.6.1.3. Длина	623
Примеры	623
11.6.1.4. Индекс	623
Примеры	624
11.6.1.5. Цвет	624
11.6.1.6. Толщина линии	624
11.6.1.7. Значение	625
11.6.1.8. Стил	625
11.6.1.9. Прозрачность	625
11.6.1.10. Отображение значений	625
11.6.1.11. Цвет фона значений	625
11.6.1.12. Прозрачность фона значений	625
11.6.1.13. Видимость	626
11.6.1.14. Формат меток на пересечении с линиями	626
11.6.1.15. Шрифт	626
11.6.1.16. Режим	626
11.6.2. События	627
11.6.2.1. ChangeValue	627
11.6.3. Функции	627
11.6.3.1. getActive	627
11.6.3.2. SetActive	627
11.7. График: точка пересечения	627
11.7.1. Свойства	627
11.7.1.1. Отображаемое имя	627
11.7.1.2. Кардинальное число	628
Значение	628
11.7.1.3. Длина	628
Примеры	628
11.7.1.4. Индекс	628
Примеры	629
11.7.1.5. Цвет	629
11.7.1.6. Стил	629
11.7.1.7. Прозрачность	629
11.7.1.8. Цвет фона значений	630
11.7.1.9. Прозрачность фона значений	630
11.7.1.10. Видимость	630
11.7.1.11. Шрифт	630
11.7.1.12. Текст	630
11.7.1.13. Признак пересечения с линией	630
11.7.1.14. Значение X	631
11.7.1.15. Значение Y	631
11.7.2. Функции	631
11.7.2.1. SetLineID	631

11.7.2.2. GetLineID	632
11.7.2.3. FirstLine	632
11.7.2.4. NextLine	632
11.7.2.5. PrevLine	632
11.7.2.6. First	632
11.7.2.7. Next	633
11.7.2.8. Prev	633
11.7.2.9. GetValue	633
11.8. График: экспорт	633
11.8.1. Свойства	634
11.8.1.1. Отображаемое имя	634
11.8.1.2. Кардинальное число	634
Значение	634
11.8.1.3. Сортировка	634
11.8.1.4. Колонка для сортировки	634
11.8.2. Функции	634
11.8.2.1. Reload	634
12. Компоненты для формирования таблиц	635
12.1. Таблица	635
12.1.1. Свойства	635
12.1.1.1. Отображаемое имя	635
12.1.1.2. Кардинальное число	635
Значение	635
12.1.1.3. Длина	635
Примеры	635
12.1.1.4. Индекс	636
Примеры	636
12.1.1.5. Модель данных	636
12.1.1.6. X и Y	636
12.1.1.7. Угол поворота	637
12.1.1.8. Масштаб	638
12.1.1.9. Видимость	638
12.1.1.10. Включено	639
12.1.1.11. Ширина и Высота	639
12.1.1.12. Высота заголовка	640
12.1.1.13. Видимость заголовка	640
12.1.1.14. Высота строки	640
12.1.1.15. Порядок вывода строк	640
12.1.1.16. Поведение выделения	641
12.1.1.17. Режим выделения	641
12.1.1.18. Следить за новой строкой	641
12.1.2. События	642
12.1.2.1. MousePress	642
12.1.2.2. MouseRelease	642
12.1.2.3. CurrentChanged	642
12.1.2.4. SelectionChanged	642
12.1.3. Функции	643
12.1.3.1. GetColumnCount	643
12.1.3.2. GetRowCount	643
12.1.3.3. GetRowDataAsJson	643
12.1.3.4. GetCellData	643
12.1.3.5. GetSelectedRows	644
12.1.3.6. SetSelectedRows	644
12.1.3.7. AddRow	644
12.1.3.8. InsertRow	644

12.1.3.9. RemoveRow	645
12.1.3.10. IsRowSelected	645
12.1.3.11. MoveColumn	645
12.1.3.12. ApplySort	646
12.1.3.13. MapColumnFromModel	646
12.1.3.14. MapColumnToModel	646
12.1.3.15. MapRowFromModel	646
12.1.3.16. MapRowToModel	647
12.1.3.17. ScrollToBottom	647
12.1.3.18. ScrollToTop	647
12.1.3.19. ScrollTo	647
12.1.3.20. GetPageHeaderTextAlignment	648
12.1.3.21. SetPageHeaderTextAlignment	648
12.1.3.22. GetPageHeaderFont	649
12.1.3.23. SetPageHeaderFont	649
12.1.3.24. GetPageHeaderText	649
12.1.3.25. SetPageHeaderText	650
12.1.3.26. GetPageFooterTextAlignment	650
12.1.3.27. SetPageFooterTextAlignment	650
12.1.3.28. GetPageFooterFont	651
12.1.3.29. SetPageFooterFont	651
12.1.3.30. GetPageFooterText	651
12.1.3.31. SetPageFooterFont	651
12.1.3.32. GetPageFooterText	652
12.1.3.33. SetPageFooterText	652
12.1.3.34. GetPageNumberFont	652
12.1.3.35. SetPageNumberFont	652
12.1.3.36. Print	653
12.1.3.37. PrintDialog	653
12.1.3.38. Save	653
12.2. Таблица: модель данных	653
12.2.1. Свойства	654
12.2.1.1. Отображаемое имя	654
12.2.1.2. Кардинальное число	654
Значение	654
12.2.1.3. Длина	654
Примеры	654
12.2.1.4. Индекс	654
Примеры	655
12.2.1.5. Заморозка	655
12.2.1.6. Максимальное количество рядов данных	655
12.2.2. События	656
12.2.2.1. DataChanged	656
12.2.2.2. ReadFinished	656
12.2.2.3. RowsInserted	656
12.2.2.4. RowsRemoved	656
12.2.3. Функции	657
12.2.3.1. GetCellData	657
12.2.3.2. GetRowDataAsJson	657
12.2.3.3. GetColumnCount	657
12.2.3.4. GetRowCount	657
12.2.3.5. SetCellData	658
12.2.3.6. SetDataAsJson	658
12.2.3.7. SetRowDataAsJson	658
12.2.3.8. AddColumn	659
12.2.3.9. AddRow	661

12.2.3.10. InsertRow	661
12.2.3.11. InsertRows	661
12.2.3.12. RemoveRow	661
12.2.3.13. RemoveRows	662
12.2.3.14. BeginReadAsync	662
12.2.3.15. CancelReadAsync	662
12.2.3.16. Clear	662
12.2.3.17. Reset	663
12.3. Таблица: столбец	663
12.3.1. Свойства	663
12.3.1.1. Отображаемое имя	663
12.3.1.2. Кардинальное число	663
Значение	663
12.3.1.3. Длина	663
Примеры	664
12.3.1.4. Индекс	664
Примеры	664
12.3.1.5. Заголовок	665
12.3.1.6. Описание	665
12.3.1.7. Идентификатор	665
12.3.1.8. Цвет фона	668
12.3.1.9. Цвет текста	668
12.3.1.10. Шрифт текста	668
12.3.1.11. Выравнивание текста	668
12.3.1.12. Выравнивание текста	668
12.3.1.13. Цвет фона заголовка	668
12.3.1.14. Цвет текста заголовка	669
12.3.1.15. Шрифт текста заголовка	669
12.3.1.16. Ширина	669
12.3.1.17. Режим изменения ширины	669
12.3.1.18. Может не иметь значения	669
12.3.1.19. Скрывать	670
12.3.1.20. Числовой формат	670
12.3.1.21. Временной формат	670
12.3.1.22. Фильтр в JSON-формате	670
Правила формирования условий фильтрации	670
12.3.1.23. Сортировать	672
12.3.1.24. Порядок сортировки	672
12.3.1.25. Приоритет сортировки	672
12.3.1.26. Разрешить перенос слов	672
12.3.2. События	673
12.3.2.1. MouseClick	673
12.3.2.2. MouseDoubleClick	673
12.3.2.3. MouseRightClick	673
12.3.2.4. MouseTitleClick	673
12.3.2.5. MouseRightTitleClick	673
12.3.2.6. WidthChanged	673
12.3.3. Функции	674
12.3.3.1. GetIndex	674
12.3.3.2. GetCellData	674
12.4. Таблица: декоратор	674
12.4.1. Свойства	674
12.4.1.1. Отображаемое имя	674
12.4.1.2. Кардинальное число	674
Значение	674
12.4.1.3. Настройки декоратора	675

12.5. Таблица: итератор	675
12.5.1. Свойства	675
12.5.1.1. Отображаемое имя	675
12.5.1.2. Кардинальное число	675
Значение	675
12.5.1.3. Индекс текущей строки	675
12.5.2. Функции	675
12.5.2.1. FirstCelectd	675
12.5.2.2. FirstCustom	676
12.5.2.3. Next	676
12.5.2.4. End	676
12.6. Таблица: устойчивый итератор	676
12.6.1. Свойства	677
12.6.1.1. Отображаемое имя	677
12.6.1.2. Кардинальное число	677
Значение	677
12.6.1.3. Длина	677
Примеры	677
12.6.1.4. Индекс	677
Примеры	678
12.6.2. Функции	678
12.6.2.1. GetData	678
12.6.2.2. GetText	679
12.6.2.3. GetRow	679
12.6.2.4. GetStatus	679
12.6.2.5. IsRowSelected	679
12.6.2.6. SetRowSelected	679
12.6.2.7. FirstRow	680
12.6.2.8. FirstSelectedRow	680
12.6.2.9. Next	680
12.6.2.10. HasNext	680
12.6.2.11. RemoveRow	680
12.6.2.12. Close	680
13. Экранные формы	682
13.1. Свойства	682
13.1.1. Отображаемое имя	682
13.1.2. X и Y	682
13.1.3. Угол поворота	683
13.1.4. Масштаб	684
13.1.5. Видимость	684
13.1.6. Включено	684
13.1.7. Всплывающая подсказка	685
13.1.8. Ширина и высота	685
13.1.9. Цвет пера	686
13.1.10. Стиль пера	686
13.1.11. Толщина пера	687
13.1.12. Цвет заливки	688
13.1.13. Стиль заливки	688
13.1.14. X окна и Y окна	690
13.1.15. Ширина окна и Высота окна	690
13.1.16. Заголовок окна	691
13.1.17. Показывать заголовок	691
13.1.18. Показывать кнопку "свернуть"	692
13.1.19. Показывать кнопку "развернуть"	692
13.1.20. Показывать кнопку "закрыть"	692

13.1.21. Поверх всех окон	693
13.1.22. Размеры окна	693
13.1.23. Стиль рамки окна	693
13.1.24. Состояние окна	694
13.1.25. Режим масштабирования	694
13.1.26. Монитор	695
13.1.27. Положение окна	696
13.1.28. Режим обработки закрытия окна	696
13.1.29. MouseX и MouseY	697
13.1.30. Активность окна	697
13.2. События	698
13.2.1. MouseMove	698
13.2.2. MouseClick	699
13.2.3. MouseRightClick	700
13.2.4. MouseDoubleClick	700
13.2.5. MousePress	701
13.2.6. MouseRelease	702
13.2.7. MouseEnter	703
13.2.8. MouseLeave	703
13.2.9. CloseRequested	704
13.2.10. Opened	704
13.2.11. Closed	704
13.2.12. WindowActivationChanged	704
13.2.13. WindowSizeChanged	705
13.2.14. WindowPosChanged	705
13.2.15. KeyPress	705
13.2.16. KeyRelease	706
13.3. Функции	707
13.3.1. ScreenX, ScreenY	707
13.3.2. Close	708
13.3.3. SaveScreenShot	708
13.3.4. Синтаксис	709
13.3.5. DoAutoSize	709
13.3.6. SetBoundingRegion	709
13.3.7. ResetBoundingRegion	710
14. Средство отладки	712
14.1. Свойства	712
14.1.1. Отображаемое имя	712
14.1.2. Кардинальное число	712
Значение	712
14.1.3. Длина	712
Примеры	712
14.1.4. Индекс	713
Примеры	713
14.2. Функции	713
14.2.1. Log	713
15. Менеджер проекта	715
15.1. Свойства	715
15.1.1. Отображаемое имя	715
15.1.2. Кардинальное число	715
Значение	715
15.2. Функции	715
15.2.1. AsyncReload	715
15.2.2. AsyncReloadFromBackup	716

15.2.3. CheckForUpdates	716
15.3. События	717
15.3.1. ReloadStarted	717
15.3.2. ReloadStarted	717
15.3.3. ReloadFailed	717
16. Среда исполнения	718
16.1. Свойства	718
16.1.1. Отображаемое имя	718
16.1.2. Кардинальное число	718
Значение	718
16.1.3. Длина	718
Примеры	718
16.1.4. Индекс	719
Примеры	719
16.1.5. Режим работы	719
Значение	719

1. Компоненты для работы с данными

1.1. Источник OPC

Компонент для связи с OPC DA источником данных. Компонент невизуальный (не отображается на форме) и виден только в области Структура объекта.



ОБРАТИТЕ ВНИМАНИЕ

Компонент недоступен при работе в ОС Linux.

1.1.1. Свойства

1.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

1.1.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

1.1.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

1.1.1.5. Хост

```
string Location
```

Сетевой адрес OPC DA источника данных. Для локального компьютера используется IP-адрес: «127.0.0.1».



ПРИМЕЧАНИЕ

Изменение этого адреса во время рантайма инициирует переподключение к новому хосту.

Примеры



ПРИМЕР

Вывести сетевой адрес источника данных в поле ввода.

```
TextEdit.Text = OpcSource.Location;
```

1.1.1.6. ProgID

```
string ProgID
```

ProgID источника данных. Для SePlatform.Data Server данный параметр содержится в конфигурационном файле SePlatform.Data Server.xml. По умолчанию параметр равен SePlatform.OPCDAServer.



ПРИМЕЧАНИЕ

Изменение ProgID во время рантайма запускает процесс переподключения к серверу с новым идентификатором.

Примеры



ПРИМЕР

Вывести ProgID источника данных в поле ввода.

```
TextEdit.Text = OpcSource.ProgId;
```

1.1.1.7. Путь

```
string Path
```

Путь до целевой ветки дерева сигналов OPC DA источника данных. Если путь не указан, то по умолчанию целевой веткой будет считаться корень дерева сигналов источника данных.



ПРИМЕЧАНИЕ

Доступна возможность переопределения пути во время рантайма.

Примеры



ПРИМЕР

Вывести путь до целевой ветки дерева сигналов источника в поле ввода.

```
TextEdit.Text = OpcSource.Path;
```

1.1.1.8. Родительский источник

Свойство для создания иерархии источников данных. Например, на экранной форме можно разместить три компонента **Источник OPC**, первый из которых будет содержать параметры подключения (свойства **Хост** и **ProgID**). Второй и третий источники лишь будут определять конкретные ветки дерева сигналов (свойство **Путь**), но ссылаться при этом на первый источник как на **Родительский источник**.

Примеры



ПРИМЕР

Записать в поле ввода сетевой адрес родительского источника.

```
TextEdit_1.Text = OpcSource_1.ParentSource.Location;
```



ПРИМЕР

Записать в поле ввода код состояния подключения родительского источника.

```
TextEdit_1.Text = String.ToString(OpcSource_1.ParentSource.ConnectionState);
```

1.1.1.9. Активность

```
bool Active
```

Активность источника OPC. С неактивными источниками не происходит обмен данными.

Примеры



ПРИМЕР

Записать текущее состояние OPC DA источника в поле ввода.

```
TextEdit_1.Text = String.ToString(OpcSource_1.Active);
```

1.1.1.10. Период обновления данных

UInt8 UpdateRate

Свойство определяет периодичность обновления данных, поступающих из источника. Измеряется в миллисекундах. К примеру, значение свойства «10000» говорит о том, что SePlatform.HMI раз в 10 секунд будет обращаться к источнику, чтобы обновить значения по всем сигналам источника, задействованным в проекте. Если значение свойства равно «0», то новые значения сигналов будут отображаться на формах SePlatform.HMI сразу же.

Примеры



ПРИМЕР

Записать значение периода обновления в текстовое поле.

```
TextEdit_1.Text = String.ToString(OpcSource_1.UpdateRate);
```

1.1.1.11. Период переподписки

UInt8 ReAdvise

Свойство определяет интервал времени, через который будет производиться переподписка на динамические сигналы источника. Измеряется в миллисекундах. К примеру, значение свойства «10000» говорит о том, что SePlatform.HMI раз в 10 секунд будет обращаться к источнику, чтобы восстановить подписку на сигналы и подписаться на появившиеся за этот интервал динамические сигналы. Если значение свойства равно «0», то переподписка не выполняется.

Примеры



ПРИМЕР

Записать значение периода переподписки в текстовое поле.

```
TextEdit_1.Text = String.ToString(OpcSource_1.ReAdvise);
```

1.1.1.12. Полный тег

string Tag

Полный путь **Источника ОРС**, формируемый путём конкатенации путей родительских источников.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

1.1.1.13. Состояние соединения

Код состояния подключения к источнику данных.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

0	отключен
1	в процессе подключения
2	подключен
3	в процессе отключения

Примеры



ПРИМЕР

Записать код состояния подключения к источнику OPC в текстовое поле.

```
TextEdit_1.Text = String.ToString(OpcSource_1.ConnectionState);
```

1.1.1.14. Тип ошибки

Код ошибки подключения к источнику OPC.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

0	ошибок нет
1	ошибка инициализации

Примеры



ПРИМЕР

Записать код ошибки подключения к источнику OPC в поле редактирования.

```
TextEdit_1.Text = String.ToString(OpcSource_1.ErrorState);
```


1.1.1.15. Строковое описание ошибки

```
string ErrorString
```

Описание ошибки подключения к источнику данных в текстовом виде.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать описание ошибки подключения к источнику OPC в поле редактирования.

```
TextEdit_1.Text = OpcSource_1.ErrorString;
```

1.1.2. События

1.1.2.1. OnConnectionStateChanged

Изменение значения свойства [Состояние соединения \(стр. 48\)](#).

Параметры

connectionState	UInt1	соответствует свойству Состояние соединения (стр. 48)
-----------------	-------	-----------------------------------------------------------------------

1.1.2.2. OnErrorStateChanged

Появление ошибок в момент изменения свойства [Состояние соединения \(стр. 48\)](#).

Параметры

errorState	UInt1	соответствует свойству Состояние ошибки
------------	-------	---------------------------------------------------------

1.2. Элемент OPC <type>

Набор компонентов для взаимодействия с OPC DA сигналами определенного типа. Компонент невизуальный (не отображается на форме) и виден только в области **Структура объекта**.



ОБРАТИТЕ ВНИМАНИЕ

Компонент недоступен при работе в ОС Linux.

1.2.1. Свойства

1.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

1.2.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

1.2.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

1.2.1.5. Источник

Ссылка на компонент **Источник OPC**, к дереву которого относится данный компонент **Элемент OPC**.

Примеры

**ПРИМЕР**

Записать в поле ввода сетевой адрес источника OPC, к которому относится сигнал.

```
TextEdit.Text = OpcItem.Source.Location;
```

1.2.1.6. Путь

```
string Path
```

Путь до сигнала относительно корня **Источника**.

Примеры



ПРИМЕР

Записать путь к OPC сигналу в поле ввода.

```
TextEdit.Text = OpcItem.Path;
```

1.2.1.7. Значение

```
variant Value
```

Значение сигнала (сигнальное свойство 2 - **Value**). Тип значения зависит от типа компонента **Элемент OPC** **<type>**. Если используется нетипизированный компонент **Элемент OPC**, то для работы с данными применяется тип **variant**.

Примеры



ПРИМЕР

Записать значение сигнала в поле редактирования при выполнении условия.

```
if(SWStatus == 0){  
    TextEdit_1.Text = String.ToString(Variant.ToInt4(OpcItem_1.Value, 0));  
}
```

1.2.1.8. Значение в виде строки

```
string ValueAsString
```

Значение сигнала в строковом формате.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать значение сигнала источника OPC в поле ввода.

```
var string = OpcItem.ValueAsString;
Object.Text = string;
```

1.2.1.9. Качество

UInt2 Quality

Качество сигнала (сигнальное свойство 3 - **Quality**).



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

«COMM_FAILURE (24)»	отсутствие соединения с источником данных
«UNCERTAIN (64)»	связь с источником данных установлена, но данные еще не получены
«BAD_CONFIG (4)»	неверно заданные параметры сигнала

Примеры



ПРИМЕР

Записать качество сигнала источника OPC в текстовое поле.

```
TextEdit_1.Text = String.ToString(OpcItem_1.Quality);
```

1.2.1.10. Метка времени

Метка времени сигнала.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать метку времени сигнала источника OPC в текстовое поле.

```
TextEdit_1.Text = String.ToString(OpcItem_1.Timestamp);
```

1.2.1.11. Полный тег

```
string Tag
```

Тег сигнала - полный путь к сигналу, начиная от корня дерева сигналов.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать полный путь к сигналу источника OPC в поле редактирования.

```
TextEdit.Text = OpcItem.Tag;
```

1.2.1.12. Оформлена ли подписка?

```
bool Advised
```

Текущее состояние подписки на сигнал.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

«True»	есть подписка на сигнал
«False»	нет подписки на сигнал

Примеры



ПРИМЕР

Записать текущее состояние подписки на сигнал источника OPC в поле ввода.

```
TextEdit_1.Text = String.ToString(OpcItem_1.Advised);
```

1.2.1.13. Единицы измерения

```
string Units
```

Единицы измерения сигнала (сигнальное свойство **100 - EUnit**).



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать единицы измерения сигнала источника OPC в текстовое поле.

```
TextField.Text = OpcItem.Units;
```

1.2.1.14. Описание

```
string Description
```

Описание сигнала (сигнальное свойство **101 - Description**).



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать описание сигнала источника OPC в текстовое поле.

```
TextField.Text = OpcItem.Description;
```

1.2.1.15. Состояние подписки

```
UInt1 AdviseState
```

Код состояния подписки на сигнал.

ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

0	не подписано (not advised)
1	выполняется процесс подписки (advising)
2	подписано (advised)
3	выполняется процесс отписки (unadvising)
4	ошибка подписки (advice error)
5	ошибка отписки (unadvice error)

Примеры

ПРИМЕР

Записать код состояния подписки на сигнал источника OPC в текстовое поле.

```
TextEdit_1.Text = String.ToString(OpcItem_1.AdviseState);
```

1.2.1.16. Строковое описание состояния подписки

```
string AdviseStateString
```

Описание состояния подписки на сигнал в текстовом виде.

ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать текст описания состояния подписки на сигнал источника OPC в текстовое поле.

```
TextField.Text = OpcItem.AdviseStateString;
```

1.2.1.17. Канонический тип данных

```
uint1 CanonicalType
```

Тип сигнала на сервере.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

1.2.2. События

1.2.2.1. OnVqtChanged

Изменение значения, качества или метки.

Параметры

value	bool	Новое значение элемента.
quality	uint4	Новое значение качества.
time	timestamp	Новое значение метки времени.

1.2.2.2. OnAdviseStateChanged

Изменение значения свойства [Состояние подписки](#).

Параметры

adviseState	UInt1	Новое состояние подписки. Соответствует свойству Состояние подписки .
--------------------	-------	---------------------------------------------------------------------------------------

1.2.2.3. OnValueChanged

Изменение значения сигнала.

Параметры

value	Новое значение элемента.
-------	--------------------------

1.2.2.4. OnValueChangedAsString

Изменение значения сигнала строкового типа.

Параметры

value	string	Новое значение в виде строки.
-------	--------	-------------------------------

1.2.2.5. OnQualityChanged

Изменение качества сигнала.

Параметры

value	UInt2	Новое значение качества.
-------	-------	--------------------------

1.2.2.6. OnTimeStampChanged

Изменение метки времени сигнала.

Параметры

time	Новое значение времени.
------	-------------------------

1.2.2.7. OnUnitsChanged

Изменение сигнального свойства 100 (EUnit).

Параметры

units	string	Новое значение единиц измерения.
-------	--------	----------------------------------

1.2.2.8. OnDescriptionChanged

Изменение значения свойства сигнала [Описание](#).

Параметры

description	string	Новое описание сигнала.
-------------	--------	-------------------------

1.2.3. Функции

1.2.3.1. TrySetValue

Записывает значение сигнала на сервер.

Синтаксис

```
bool TrySetValue (Value)
```

Параметры

value	variant
-------	---------

Возвращаемое значение

«true»	значение успешно записано на сервер
«false»	не удалось записать значение

Примеры



ПРИМЕР

Попытаться присвоить значение сигналу и вывести результат в текстовое поле.

```
Result: var = Opcitem_1.TrySetValue("Строка 1");  
TextEdit_1.Text = String.ToString(Result);
```

1.3. Источник AP

Компонент для связи с источником данных по протоколу TCP. Компонент невизуальный (не отображается на форме) и виден только в области **Структура** объекта.

1.3.1. Функции

1.3.1.1. Ask

Квитировать событие.

Синтаксис

```
void Ack(string Source, string conditionName, string comment, string acknowledgedId, uint8
activeTime, uint4 cookie)
```

Параметры

Source	Источник события
conditionName	Имя условия события.
comment	Комментарий к квитированию.
acknowledgerId	Идентификатор пользователя, квитирующего событие.
activeTime	Активное время события.
cookie	Куки, ассоциированные с событием.

Примеры

ПРИМЕР

Квитировать событие

```
ApSource_1.Ack("Thermometer1", "HighTemperature", "Температура проверена и
стабилизирована", "user123", 162030, 123456);
```

1.3.2. Свойства

1.3.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.3.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

1.3.2.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

1.3.2.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

1.3.2.5. Хост

string Location

Сетевой адрес источника данных по протоколу TCP. Для локального компьютера используется IP-адрес: «127.0.0.1».

Примеры

**ПРИМЕР**

Вывести сетевой адрес источника данных в поле ввода.

```
TextEdit.Text = ApSource.Location;
```

1.3.2.6. Порт

UInt2 Port

Номер порта для подключения к источнику данных. По умолчанию номер «4388».

**ПРИМЕР**

Вывести порт источника данных в поле ввода.

```
TextEdit_1.Text = String.ToString(ApSource_1.Port);
```

1.3.2.7. Порт истории

UInt2 HistoryPort

Порт для доступа к историческим данным. По умолчанию номер «4950».

1.3.2.8. Путь

```
string Path
```

Путь до целевой ветки дерева сигналов источника данных по протоколу TCP. Если путь не указан, то по умолчанию целевой веткой будет считаться корень дерева сигналов источника данных.



ПРИМЕЧАНИЕ

Доступна возможность переопределения пути во время рантайма.

Примеры



ПРИМЕР

Вывести путь до целевой ветки дерева сигналов источника в поле ввода.

```
TextEdit.Text = ApSource.Path;
```

1.3.2.9. Источник учетных данных

```
Credentials
```

Источник учетных данных, используемый для подключения к защищенному источнику. Указывается в виде ссылки на объект типа **Учетные данные AP**, существующий в проекте.

1.3.2.10. Родительский источник

Свойство для создания иерархии источников данных. Например, на экранной форме можно разместить три компонента **Источник AP**, первый из которых будет содержать параметры подключения (свойства **Хост** и **Порт**). Второй и третий источники лишь будут определять конкретные ветки дерева сигналов (свойство **Путь**), но ссылаться при этом на первый источник как на **Родительский источник**.

Примеры



ПРИМЕР

Записать в поле ввода сетевой адрес родительского источника.

```
TextEdit.Text = ApSource.ParentSource.Location;
```



ПРИМЕР

Записать в поле ввода код состояния подключения родительского источника.

```
TextEdit_1.Text = String.ToString(ApSource_1.ParentSource.ConnectionState);
```

1.3.2.11. Таймаут операций с БД

uint4 Timeout

Максимальное время в миллисекундах для выполнения операций с базой данных. При превышении этого времени операция прерывается.

Примеры



ПРИМЕР

Установить таймаут операций в 5 секунд

```
ApSource_1.Timeout = 5000;
```

1.3.2.12. Таймаут установления соединения с БД

uint4 ConnectionEstablishTimeout

Время ожидания в миллисекундах для установления соединения с базой данных. Если соединение не устанавливается в течение этого времени, процесс подключения прерывается.

Примеры



ПРИМЕР

Установить таймаут соединения с БД в 10 секунд

```
ApSource_1.ConnectionEstablishTimeout = 10000;
```

1.3.2.13. Максимальный размера пула буферов

uint4 MaxBufferPoolSize

Максимальный размер пула буферов в байтах, используемого для временного хранения данных перед их записью в SePlatform.Historian.

1.3.2.14. Общее соединение

bool ShareConnection

Использовать ли общее соединение для всех источников данных, имеющих одинаковый адрес подключения (**Хост**, **Порт**). Если этот параметр не установлен («false»), то для каждого источника данных будет

устанавливаться отдельное соединение. По умолчанию используется общее соединение. Это свойство актуально только в случаях, когда не указан родительский источник данных.

1.3.2.15. Активность

```
bool Active
```

Активность источника AP. С неактивными источниками не происходит обмен данными.

Примеры



ПРИМЕР

Записать текущее состояние источника AP в поле ввода.

```
TextEdit_1.Text = String.ToString(ApSource_1.Active);
```

1.3.2.16. Период переподписки

```
uint8 ReAdvise
```

Интервал времени в миллисекундах, после которого будут предприниматься попытки переподписки на сигналы, отсутствующие в сервере. Значение «0» - переподписка не выполняется.

1.3.2.17. Имя клиента

```
string ClientDisplayName
```

Имя клиента, используемое для удобного отображения подключенного клиента к источнику AP.

Примеры



ПРИМЕР

Установить имя клиента «Оператор»;

```
ApSource.ClientDisplayName = "Оператор";
```

1.3.2.18. Идентификатор клиента

```
string ClientId
```

Идентификатор клиента, используемый для идентификации в рамках подключения к источнику AP. Пустое значение указывает на анонимного клиента.

Примеры

ПРИМЕР

Установить уникальный идентификатор клиента

```
ApSource_1.ClientId = "client12345";
```

1.3.2.19. Полный тег

```
string Tag
```

Полный путь **Источника AP**, формируемый путём конкатенации путей родительских источников.

ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

1.3.2.20. Состояние соединения

```
UInt1 ConnectionState
```

Код состояния подключения к источнику данных.

ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

0	отключен
1	в процессе подключения
2	подключен
3	в процессе отключения

Примеры



ПРИМЕР

Записать код состояния подключения к источнику AP в текстовое поле.

```
TextEdit_1.Text = String.ToString(ApSource_1.ConnectionState);
```

1.3.2.21. Тип ошибки

Код ошибки подключения к источнику данных.

```
UInt1 ErrorState
```



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

0	ошибок нет
1	ошибка инициализации

Примеры



ПРИМЕР

Записать код ошибки подключения к источнику AP в поле редактирования.

```
TextEdit_1.Text = String.ToString(ApSource_1.ErrorState);
```

1.3.2.22. Строковое описание ошибки

```
string ErrorString
```

Описание ошибки подключения к источнику данных в текстовом виде.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать описание ошибки подключения к источнику AP в поле редактирования.

```
TextEdit.Text = ApSource.ErrorString;
```

1.3.3. События

1.3.3.1. OnConnectionStateChanged

Изменение значения свойства [Состояние соединения](#) (стр. 66).

Параметры

connectionState	UInt1	соответствует свойству Состояние соединения (стр. 66)
-----------------	-------	-----------------------------------------------------------------------

1.3.3.2. OnErrorStateChanged

Появление ошибок в момент изменения свойства [Состояние соединения](#) (стр. 66).

Параметры

errorState	UInt1	соответствует свойству Состояние ошибки
------------	-------	---------------------------------------------------------

1.4. Элемент AP <type>

Набор компонентов для взаимодействия с сигналами определенного типа по протоколу TCP. Компонент не визуальный (не отображается на форме) и виден только в области **Структура** объекта.

1.4.1. Функции

1.4.1.1. TrySetValue

Записывает значение сигнала на сервер.

Синтаксис

```
bool TrySetValue (Value)
```

Параметры

value	variant
-------	---------

Возвращаемое значение

«true»	значение успешно записано на сервер
«false»	не удалось записать значение

Примеры



ПРИМЕР

Попытаться присвоить значение сигналу и вывести результат в текстовое поле.

```
Result: var = ApItem_1.TrySetValue("Строка 1");
TextEdit_1.Text = String.ToString(Result);
```

1.4.2. Свойства

1.4.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.4.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

1.4.2.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

1.4.2.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

1.4.2.5. Источник

Ссылка на компонент **Источник АР**, к дереву которого относится данный компонент **Элемент АР**.

Примеры

**ПРИМЕР**

Записать в поле ввода сетевой адрес источника, к которому относится сигнал.

```
TextEdit.Text = ApItem.Source.Location;
```

1.4.2.6. Путь

```
string Path
```

Путь до сигнала относительно корня **Источника**.

Примеры

**ПРИМЕР**

Записать путь к сигналу источника в поле ввода.

```
TextEdit.Text = ApItem.Path;
```

1.4.2.7. Значение

```
variant Value
```

Значение сигнала (сигнальное свойство 2 - **Value**). Тип значения зависит от типа компонента **Элемент АР** **<type>**. Если используется нетипизированный компонент **Элемент АР**, то для работы с данными применяется тип **variant**.

Примеры



ПРИМЕР

Увеличить значение сигнала при выполнении условия.

```
if(SWStatus == 1){
    ApItem.Value += 1;
}
```



ПРИМЕР

Записать значение сигнала в поле редактирования при выполнении условия.

```
if(SWStatus == 0){
    TextEdit_1.Text = String.ToString(Variant.ToInt4(ApItem_1.Value, 0));
}
```

1.4.2.8. Значение в виде строки

```
string ValueAsString
```

Значение сигнала в строковом формате.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать значение сигнала источника AP в поле ввода.

```
var string = ApItem.ValueAsString;
Object.Text = string;
```

1.4.2.9. Качество

```
UInt2 Quality
```

Качество сигнала (сигнальное свойство 3 - *Quality*).



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

«COMM_FAILURE (24)»	отсутствие соединения с источником данных
«UNCERTAIN (64)»	связь с источником данных установлена, но данные еще не получены
«BAD_CONFIG (4)»	неверно заданные параметры сигнала

Примеры



ПРИМЕР

Записать качество сигнала источника AP в текстовое поле.

```
TextEdit_1.Text = String.ToString(ApItem_1.Quality);
```

1.4.2.10. Метка времени

Метка времени сигнала.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать метку времени сигнала источника AP в текстовое поле.

```
TextEdit_1.Text = String.ToString(ApItem_1.TimeStamp);
```

1.4.2.11. Полный тег

```
string Tag
```

Тег сигнала - полный путь к сигналу, начиная от корня дерева сигналов.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать полный путь к сигналу источника AP в поле редактирования.

```
TextEdit.Text = ApItem.Tag;
```

1.4.2.12. Флаг ведения истории

```
bool Historizing
```

Активирована ли функция записи истории для сигнала на сервере.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

«true»	История сигнала записывается.
«false»	История сигнала не записывается.

1.4.2.13. Канонический тип данных

```
uint1 CanonicalType
```

Тип сигнала на сервере.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

1.4.2.14. Оформлена ли подписка?

```
bool Advised
```

Текущее состояние подписки на сигнал.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

«True»	есть подписка на сигнал
«False»	нет подписки на сигнал

Примеры



ПРИМЕР

Записать текущее состояние подписки на сигнал источника AP в поле ввода.

```
TextEdit_1.Text = String.ToString(ApItem_1.Advised);
```

1.4.2.15. Единицы измерения

```
string Units
```

Единицы измерения сигнала (сигнальное свойство **100** - **EUnit**).



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать единицы измерения сигнала источника AP в текстовое поле.

```
TextField.Text = ApItem.Units;
```

1.4.2.16. Описание

```
string Description
```

Описание сигнала (сигнальное свойство **101** - **Description**).



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать описание сигнала источника AP в текстовое поле.

```
TextField.Text = ApItem.Description;
```

1.4.2.17. Состояние подписки

UInt1 AdviseState

Код состояния подписки на сигнал.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

0	не подписано (not advised)
1	выполняется процесс подписки (advising)
2	подписано (advised)
3	выполняется процесс отписки (unadvising)
4	ошибка подписки (advice error)
5	ошибка отписки (unadvice error)

Примеры



ПРИМЕР

Записать код состояния подписки на сигнал источника AP в текстовое поле.

```
TextEdit_1.Text = String.ToString(ApItem_1.AdviseState);
```

1.4.2.18. Строковое описание состояния подписки

string AdviseStateString

Описание состояния подписки на сигнал в текстовом виде.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать текст описания состояния подписки на сигнал источника AP в текстовое поле.

```
TextField.Text = ApItem.AdviseStateString;
```

1.4.3. События

1.4.3.1. OnVqtChanged

Изменение значения, качества или метки времени.

Параметры

value	bool	Новое значение элемента.
quality	uint4	Новое значение качества.
time	timestamp	Новое значение метки времени.

1.4.3.2. OnAdviseStateChanged

Изменение значения свойства [Состояние подписки](#).

Параметры

adviseState	UInt1	Новое состояние подписки. Соответствует свойству Состояние подписки .
-------------	-------	---------------------------------------------------------------------------------------

1.4.3.3. OnValueChanged

Изменение значения сигнала.

Параметры

value	Новое значение элемента.
-------	--------------------------

1.4.3.4. OnValueChangedAsString

Изменение значения сигнала строкового типа.

Параметры

value	string	Новое значение в виде строки.
-------	--------	-------------------------------

1.4.3.5. OnQualityChanged

Изменение качества сигнала.

Параметры

value	UInt2	Новое значение качества.
-------	-------	--------------------------

1.4.3.6. OnTimeStampChanged

Изменение метки времени сигнала.

Параметры

time	Новое значение времени.
------	-------------------------

1.4.3.7. OnUnitsChanged

Изменение сигнального свойства 100 (EUnit).

Параметры

units	string	Новое значение единиц измерения.
-------	--------	----------------------------------

1.4.3.8. OnDescriptionChanged

Изменение значения свойства сигнала [Описание](#).

Параметры

description	string	Новое описание сигнала.
-------------	--------	-------------------------

1.4.3.9. OnHistorizingChanged

Изменение значения свойства [Флаг ведения истории](#).

Параметры

historizing	bool	Новое состояние флага ведения истории.
-------------	------	----------------------------------------

1.5. Учетные данные AP

Компонент используется для подключения к защищенному паролем источнику данных в проекте SePlatform.HMI.

В качестве источника может использоваться, к примеру, SePlatform.Data Server.

1.5.1. Свойства

1.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	Одиночный объект.
> «1»	Массив соответствующей размерности.

1.5.1.3. Путь к файлу с учетными данными

```
string FilePath
```

Путь к файлу, в котором сохраняются с помощью функции [Save \(стр. 81\)](#) учетные данные для подключения к защищенному источнику.



ПРИМЕЧАНИЕ

Пользователь самостоятельно создает файл и выбирает расширение файла на свое усмотрение (например, .txt, .doc, .tex, .text и другие). Также допустимо использование файла без расширения.

Если путь к файлу указан, пароль будет автоматически подгружен при запуске проекта в рантайме, позволяя осуществить автоматическую аутентификацию. Если путь не указан, пользователю придется вводить пароль вручную через интерфейсный элемент, например, текстовое поле, при каждом запуске проекта в рантайме.

Примеры



ПРИМЕР

Вывести в текстовое поле заданный путь к файлу с учетными данными.

```
Text_1.Text = ApCredentials_1.FilePath;
```

1.5.1.4. Присвоены ли учетные данные

Доступно только для чтения в режиме рантайма.

```
bool Assigned
```

Указывает на то, были ли предоставлены учетные данные компоненту **Учетные данные AP** в проекте SePlatform.HMI для подключения к защищенному источнику данных.

Значение

«true»	Учетные данные предоставлены.
«false»	Учетные данные не предоставлены.

Примеры



ПРИМЕР

Проверить, предоставлены ли учетные данные компоненту **Учетные данные AP** для подключения к защищенному источнику. Результат вывести в текстовое поле.

```
Text_1.Text = String.ToString(ApCredentials_1.Assigned);
```

1.5.2. Функции

1.5.2.1. SetPassword

Позволяет указать пароль для подключения к защищенному источнику данных.

Синтаксис

```
void SetPassword(string Password)
```


Параметры

Password	Строковый параметр, содержащий пароль.
----------	----------------------------------------

Примеры



ПРИМЕР

Указать пароль доступа «Qwerty1234-» для подключения к защищенному источнику данных.

```
ApCredentials_1.SetPassword("Qwerty1234-");
```

1.5.2.2. Save

Сохраняет пароль доступа в зашифрованном виде (хэш пароля) в файл, который указывается в свойстве [Путь к файлу с учетными данными \(стр. 79\)](#).

Синтаксис

```
bool Save()
```

Значение

«true»	Сохранение учетных данных прошло успешно.
«false»	Свойство Путь к файлу с учетными данными (стр. 79) не заполнено или сохранение не удалось (например, файла по указанному пути не существует).



ПРИМЕР

Сохранить пароль для подключения к защищенному источнику данных и вывести результат в текстовое поле.

```
Text_1 = "Save: " + (String.ToString(ApCredentials_1.Save()));
```

1.5.2.3. Load

Загружает пароль доступа из файла, указанного в свойстве [Путь к файлу с учетными данными \(стр. 79\)](#).

Синтаксис

```
bool Load()
```

Значение

«true»	Загрузка пароля прошла успешно.
«false»	Свойство Путь к файлу с учетными данными (стр. 79) не заполнено или загрузка пароля не удалась (например, файл с учетными данными отсутствует).



ПРИМЕР

Загрузить пароль к защищенному источнику данных из файла и вывести результат в текстовое поле.

```
Text_1 = "Load: " + (String.ToString(ApCredentials_1.Load()));
```

1.5.2.4. GetLastError

Возвращает текстовое описание последней возникшей ошибки при выполнении операций с учетными данными. Если ошибок не было, функция вернет пустую строку.

Синтаксис

```
string GetLastError()
```



ПРИМЕР

Вывести в поле ввода текстовое описание ошибок, возникших при выполнении последних операций с учетными данными.

```
LineEdit_1 = ApCredentials_1.GetLastError();
```

1.6. Браузер источника AP

Компонент, выполняющий браузеринг дерева сигналов источника AP.

1.6.1. Функции

1.6.1.1. GetChildren

При вызове этой функции компонент **Браузер источника АР** выполняет браузинг узлов дерева сигналов источника данных, начиная с указанного родительского узла. Если параметр «parentNode» не задан (равен пустой строке), то браузинг начинается с корня дерева.

Синтаксис

```
void GetChildren(string parentNode)
```

Параметры

parentNode	Строка, представляющая путь к родительскому узлу.
------------	---------------------------------------------------

Примеры



ПРИМЕР

Выполнить браузинг узлов дерева сигналов, начиная с корня.

```
ApSourceBrowser_1.GetChildren(" ");
```



ПРИМЕР

Выполнить браузинг узлов дерева сигналов, начиная с пути папка1.общие параметры.сигналы.

```
ApSourceBrowser_1.GetChildren("папка1.общие параметры.сигналы");
```

1.6.1.2. GetSourceNodes

Получает информацию о поддереве узла источника данных, основываясь на заданном фильтре.

Параметр «filter» представляет собой строку с выражением в формате wildcard filter, где символы «*» и «?» используются для фильтрации узлов.

Синтаксис

```
void GetSourceNodes(string filter)
```

Параметры

filter	<p>Строка, содержащая выражение в формате wildcard filter для фильтрации узлов. Пример фильтрации:</p> <ul style="list-style-type: none"> ➤ «server_*» - этот фильтр выберет все узлы, начинающиеся с «server_», например, «server_1», «server_data», «server_backup», и т. д. ➤ «param?» - этот фильтр выберет все узлы, где после слова «param» идет ровно один любой символ, например, «param1», «param2», «paramA», но не «param», так как в этом случае после «param» нет символа.
--------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Примеры



ПРИМЕР

Отфильтровать узлы источника данных по имени, оставляя только те, которые начинаются с «sensor».

```
// Фильтр для поиска узлов, начинающихся с "sensor"
filter: string = "sensor*";
// Вызов функции GetSourceNodes() с передачей фильтра
ApSourceBrowser_1.GetSourceNodes(filter);
```

1.6.1.3. GetChildrenWithRequestId

При вызове этой функции компонент **Браузер источника AP** создает запрос к источнику данных для получения узлов дерева сигналов по указанному родительскому узлу.

Результаты запроса (дочерние узлы) автоматически прикрепляются к узлу с идентификатором запроса «requestId» в графическом компоненте **Дерево** (или другой структуре данных), который был предварительно создан с использованием метода [AddItem\(\)](#).

Синтаксис

```
void GetChildrenWithRequestId(string parentNode, string requestId)
```

Параметры

«parentNode»	Строка, представляющая родительский узел, для которого будет выполнен браузер.
«requestId»	Идентификатор запроса, который соответствует родительскому узлу, к которому должен быть прикреплен результат браузера.

Примеры



ПРИМЕР

Создать корневой элемент «Модули» в компоненте **Дерево** и загрузить дочерние элементы для узла «Модули».

```
x: string = Tree_1.AddItem("", "Модули");
```

Вызываем метод **GetChildrenWithRequestId()** для загрузки дочерних элементов для узла «Модули» с использованием идентификатора «x»:

```
ApSourceBrowser_1.GetChildrenWithRequestId("Модули", x);
```

1.6.2. Свойства

1.6.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.6.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	Одиночный объект.
> «1»	Массив соответствующей размерности.

1.6.2.3. Источник AP

Source

Ссылка на компонент **Источник AP**, существующий в проекте.

1.6.2.4. Вывод описания предпочтительнее имени

bool DescriptionPreffered

Выводить описание узлов вместо их имен при отображении информации в дереве узлов.

Значение

«true»	Вывод описания узлов вместо их имен.
«false»	Вывод имени узлов вместо их описания.

Примеры



ПРИМЕР

Выводить описание узлов вместо имени.

```
ApSourceBrowser_1.DescriptionPreffered = true;
```

1.7. Запрос алармов

Компонент выполняет роль провайдера данных и позволяет получать с сервера оперативные и исторические события по протоколу TCP. Компонент не визуальный (не отображается на форме) и виден только в области Структура объекта.

1.7.1. Свойства

1.7.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.7.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

1.7.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

1.7.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){
  TextEdit_1.Text = "Включен режим 1";}
else
  {TextEdit_1.Text = "Включен режим 2";}
```

1.7.1.5. Источник

Сервер, поставляющий оперативные и исторические события. Указывается в виде ссылки на объект типа **Источник АР**, существующий в проекте.

1.7.1.6. Режим работы

UInt1 Mode

Устанавливает режим работы компонента в зависимости от вида запрашиваемых событий.

Значение

1	Список активных условий
2	Журнал оперативных событий
3	Журнал исторических событий

1.7.1.7. Начало исторического интервала, Конец исторического интервала

**ПРИМЕЧАНИЕ**

Свойство используется только при запросе исторических данных (когда в свойстве **Режим работы** ([стр. 88](#)) указано значение «3»).

timestamp HistoricalRangeStart

timestamp HistoricalRangeEnd

Начальное и конечное значение интервала запроса исторических данных от источника. Перед указанием значения переведите ваше строковое значение даты в метку времени формата uint8, а затем полученное

значение переведите в формат UTC. Для перевода вы можете использовать функции **Parse** и **ToUtc** языка SePlatform.Om.

**ПРИМЕЧАНИЕ**

Начальной или конечной точкой интервала запроса может быть «0». Это значит, что данные будут запрошены за бесконечный интервал в прошлом.

Пример

**ПРИМЕР**

Запросить исторические события с «22.07.2022 20:00:25» по «25.07.2022 21:17:25».

```
DateStart: uint8 = DateTime.Pars ("22.07.2022 20:00:25");
DateEnd: uint8 = DateTime.Parse ("25.07.2022 21:17:25");
Events.TableModelEvents.QuerySource1.HistoricalRangeStart = DateTime.ToUtc
(DateStart);
Events.TableModelEvents.QuerySource1.HistoricalRangeEnd = DateTime.ToUtc(DateEnd);
Events.TableModelEvents.QuerySource1.Reload();
Events.TableModelEvents.BeginReadAsync();
```

1.7.1.8. Фильтр в JSON-формате

string Filter

Условия фильтрации запрашиваемых от источника (сервера, генерирующего события) оперативных данных. Задаются в виде строки - объекта JSON.

Фильтрация возможна только для столбцов с идентификаторами (переменными): **severity**, **message**, **source**, **area**, **acked**.

Условие фильтрации представляет собой одиночный ({"ключ": "значение"}) или составной (состоящий из нескольких объектов) объект JSON.

Составной объект формируется с помощью:

- логических операций:
 - конъюнкция - "and" или "&"
 - дизъюнкция - "or" или "|"
 - инверсия - "not" или "!"
- операций сравнения:
 - больше - ">"
 - больше либо равно - ">="
 - меньше - "<"
 - меньше либо равно - "<="
 - соответствует - "=="
 - не соответствует - "!="

В зависимости от используемой операции составной объект JSON может иметь вид:

- {"and": [ARRAY]}
- {"or": [ARRAY]}
- {"not": {OBJECT}}
- {"операция сравнения": { VARIABLE }} (операцию сравнения "==" можно не указывать, т.е. вид объекта - { VARIABLE })

где:

- VARIABLE - переменная. Обозначается структурой <имя переменной>: <значение переменной>
- OBJECT - переменная или составной объект JSON любого из указанных выше форматов
- ARRAY - массив. Может состоять из переменных и составных объектов JSON (любого формата) в различных комбинациях

Примеры



ПРИМЕР

Запросить события с:

- важностью (столбец severity) больше «10»;
- сообщением (столбец message), содержащим текст «поставлен» или с именем источника (столбец source), не соответствующим «TU_212».

```
{
  "AND": [
    { ">": { "severity": 10 } },
    {
      "OR": [
        { "message": "*поставлен*" },
        { "NOT": { "source": "TU_212" } }
      ]
    }
  ]
}
```



ПРИМЕР

Запросить события с важностью «667».

```
{"==":{"severity":667}}
```

1.7.1.9. Фильтр исторических данных в JSON-формате (HistoricalFilter)

```
string HistoricalFilter
```

Условия фильтрации запрашиваемых от источника исторических данных. Задаются в виде строки - объекта JSON. Правила формирования условий фильтрации аналогичны правилам при запросе оперативных данных ([стр. 89](#)).



ПРИМЕЧАНИЕ

Фильтр можно установить для всех данных, загружаемых в историческом режиме (когда в свойстве **Режим работы** ([стр. 88](#)) указано значение «3»), а также для исторических данных, загружаемых в оперативном режиме (когда в свойстве **Режим работы** ([стр. 89](#)) указано значение «2» или «1», а в свойстве **Интервал предзагрузки истории** ([стр. 92](#)) задан интервал предзагрузки данных).

1.7.1.10. Активность

bool Active

Включает/выключает активность компонента.



ПРИМЕЧАНИЕ

Неактивный компонент не запрашивает события.

Значения

«true»	Компонент активен и может запрашивать события.
«false»	Компонент неактивен.

1.7.1.11. Удалять квитированные и деактивированные события

bool RemoveAkedInactive

Разрешает/запрещает удаление из списка событий, получаемых от источника (сервера, поставляющего оперативные события) квитированные и при этом деактивированные события.



ПРИМЕЧАНИЕ

Свойство работает только в режиме ([стр. 88](#)) журнала оперативных событий.

Значения

«true»	Квитированные и деактивированные события удаляются из списка.
«false»	Квитированные и деактивированные события остаются в списке.

1.7.1.12. Запрашивать список активных событий при подключении к источнику

bool RefreshAtStart

Включает запрос активных и несквитированных событий с источника данных ([стр. 88](#)) при каждом подключении.

Значения

«true»	Запрос активных и несквитированных событий при подключении к серверу включен.
«false»	Запрос активных и несквитированных событий при подключении к серверу выключен.

1.7.1.13. Интервал предзагрузки истории

UInt4 HistoryPreloadInterval

Интервал (в секундах) подгрузки исторических данных в таблицу оперативных данных. Интервал начинается указанное количество секунд назад и заканчивается текущим моментом.

1.7.1.14. Расширяющий интервал



ПРИМЕЧАНИЕ

Свойство используется только при запросе исторических данных (когда в свойстве **Режим работы** ([стр. 88](#)) указано значение «3»).

UInt4 HistoryExpandInterval

Отрезок времени (в часах), на который сдвигается время, указанное в свойстве **Конец исторического интервала** ([стр. 88](#)). За счет такого сдвига можно расширить исходный запрашиваемый интервал (указанный в свойствах **Начало исторического интервала** ([стр. 88](#)) и **Конец исторического интервала** ([стр. 88](#))).

Если такое расширение установлено, с сервера запрашиваются также события, у которых время генерации больше времени, указанного в свойстве **Конец исторического интервала** ([стр. 88](#)). Из полученных событий отбрасываются события, время срабатывания которых не входит в исходный запрашиваемый интервал.

1.7.1.15. Хронология исторических событий



ПРИМЕЧАНИЕ

Свойство используется только при запросе исторических данных (когда в свойстве [Режим работы](#) (стр. 88) указано значение «3»).

UInt4 HistoricalEventsChronology

Устанавливает хронологию, в которой будут запрашиваться исторические события за интервал, указанный в свойствах [Начало исторического интервала](#) (стр. 88) и [Конец исторического интервала](#) (стр. 88).

Значение

« 0 »	По времени генерации. Хронология выстраивается с учетом времени генерации уведомлений о событиях на сервере.
« 1 »	По времени срабатывания. Хронология выстраивается с учетом времени активации условий на технологическом объекте.

1.7.2. Функции

1.7.2.1. Reload

Применяет все заданные свойства объекта типа [Запрос алармов](#). По запросу потребителя (например, компонента после [Таблица: модель данных](#) из юнита [Таблицы](#)) функция обновляет данные с источника - сервера, поставляющего оперативные и исторические данные.

Синтаксис

```
void Reload()
```

Примеры



ПРИМЕР

Запросить данные с источника с учетом фильтра, заданного в текстовом поле.

```
ApAlarmQuery_1.JsonFilter = TextEdit_1.Text;
ApAlarmQuery_1.Reload();
```

1.8. Запрос значений элемента AP

1.8.1. Свойства

1.8.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.8.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

1.8.1.3. Элемент AP

Ссылка на Элемент AP.

1.8.1.4. Режим работы

Uint4 Mode

Режим работы компонента в зависимости от вида запрашиваемых значений.

Значение

«1»	исторические значения
«2»	оперативные значения

1.8.1.5. Начало исторического интервала

timestamp HistoricalRangeStart

Начальное значение интервала запроса исторических данных от источника.

1.8.1.6. Конец исторического интервала

```
timestamp HistoricalRangeEnd
```

Конечное значение интервала запроса исторических данных от источника.

1.8.1.7. Активность

```
bool Active
```

Активация компонента. С неактивным компонентом не происходит обмен данными.

1.8.1.8. Оперативный интервал

```
UInt4 OperativeInterval
```

Общий период времени (в секундах), за который предварительно читаются и подготавливаются к отображению данные.

1.8.1.9. Размер оперативного буфера

```
UInt4 OperativeBufferSize
```

Период времени (в секундах), отображаемый на трендовом поле.

1.8.1.10. Предзагрузка исторических данных

```
bool PreloadHistoricalData
```

Включение/отключение подгрузки исторических данных на графики перед отображением новых значений.

1.8.2. Функции

1.8.2.1. Reload

Применяет все заданные свойства у объекта типа **Запрос значений элемента AP**. По запросу потребителя (например, компонента **Таблица: модель данных** из юнита **Таблицы**) функция обновляет данные с источника данных, поставляющего оперативные и исторические данные.

Синтаксис

```
void Reload()
```

Примеры



ПРИМЕР

Запросить оперативные данные с источника.

```
ApItemValueQuery_1.Mode=2;  
ApItemValueQuery_1.Reload();
```

1.9. Запрос значений множества сигналов AP

1.9.1. Функции

1.9.1.1. Reload

Применяет все заданные свойства у объекта типа **Запрос значений множества сигналов AP**. По запросу потребителя (например, компонента **Таблица: модель данных** из юнита **Таблицы**) функция обновляет данные с источника данных, поставляющего исторические данные.

Синтаксис

```
void Reload()
```

Примеры



ПРИМЕР

Запросить значения множества сигналов с источника.

```
ApMultiItemValueQuery_1.Reload( );
```


1.9.2. Свойства

1.9.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.9.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

1.9.2.3. Источник AP

Ссылка на компонент **Источник AP**.

1.9.2.4. Начало исторического интервала

```
timestamp HistoricalRangeStart
```

Начальное значение интервала запроса исторических данных от источника.

1.9.2.5. Конец исторического интервала

```
timestamp HistoricalRangeEnd
```

Конечное значение интервала запроса исторических данных от источника.

1.9.2.6. Список сигналов

```
string signals
```

Свойство предназначено для указания списка сигналов источника данных, по которым нужно получить данные. Формат задания - JSON.

Передаваемая JSON-структура должна иметь следующий формат:

```
{
  "items": [
    {
      "tag": "<TAG>",
      "value": "<COLUMN-NAME>",
      "quality": "<COLUMN-NAME>",
      "timestamp": "<COLUMN-NAME>",
    },
    ...
  ]
}
```

Где:

- «tag»- тег сигнала на сервере (полное имя сигнала, формирующееся из имен узлов дерева сигналов, последовательно включающих данный сигнал).
- «value»- идентификатор столбца таблицы, в который будет записываться значение сигнала.
- «quality»- идентификатор столбца таблицы, в который будет записываться качество сигнала (необязательный параметр).
- «timestamp»- идентификатор столбца таблицы, в который будет записываться метка времени сигнала (необязательный параметр).

Примеры



ПРИМЕР

Настройте компонент **Запрос значений множества сигналов AP** так, чтобы значения сигналов с тегами «Deviation.s_deviation4» и «Level.s_level4» были записаны в столбцы с идентификаторами «Value_1» и «Value_2».

В свойство **Список сигналов** передадим JSON-структуру, которая описывает сигналы и их ассоциированные столбцы в выходном наборе данных:

```
{
  "items": [
    {
      "tag": "Deviation.s_deviation4",
      "value": "Value_1"
    },
    {
      "tag": "Level.s_level4",
      "value": "Value_2"
    }
  ]
}
```

1.10. Очередь активных алармов

Компонент позволяет создать очередь активных несквитированных событий (работает только в оперативном режиме). Компонент невизуальный (не отображается на форме) и виден только в области **Структура объекта**.

1.10.1. Свойства

1.10.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

1.10.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

1.10.1.3. Активность

Свойство показывает, есть ли события в очереди.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

bool Active

Значения

«true»	В очереди есть события.
«false»	В очереди нет событий.

1.10.1.4. Запрос алармов

Источник событий для очереди. Указывается в виде ссылки на объект типа **Запрос алармов**, существующий в проекте.

1.10.1.5. Фильтр в JSON-формате

string Filter

Условия фильтрации запрашиваемых от источника (объекта типа **Запрос алармов**) данных. Условия фильтрации задаются в виде строки - объекта JSON.

Фильтрация возможна только для столбцов с идентификаторами (переменными): **severity, message, source, area, acked**.

Условие фильтрации представляет собой одиночный ({"ключ": "значение"}) или составной (состоящий из нескольких объектов) объект JSON.

Составной объект формируется с помощью:

- логических операций:
 - конъюнкция - "and" или "&"
 - дизъюнкция - "or" или "|"
 - инверсия - "not" или "!"
- операций сравнения:
 - больше - ">"
 - больше либо равно - ">="
 - меньше - "<"
 - меньше либо равно - "<="
 - соответствует - "=="
 - не соответствует - "!="

В зависимости от используемой операции составной объект JSON может иметь вид:

- {"and": [ARRAY]}
- {"or": [ARRAY]}
- {"not": {OBJECT}}
- {"операция сравнения": {VARIABLE}} (операцию сравнения "==" можно не указывать, т.е. вид объекта - {VARIABLE})

где:

- VARIABLE - переменная. Обозначается структурой <имя переменной>: <значение переменной>
- OBJECT - переменная или составной объект JSON любого из указанных выше форматов
- ARRAY - массив. Может состоять из переменных и составных объектов JSON (любого формата) в различных комбинациях

Примеры



ПРИМЕР

Запросить события с:

- важностью (столбец `severity`) больше «10»;
- сообщением (столбец `message`), содержащим текст «поставлен» или с именем источника (столбец `source`), не соответствующим «TU_212».

```
{
  "AND": [
    { ">": { "severity": 10 } },
    {
      "OR": [
        { "message": "*поставлен*" },
        { "NOT": { "source": "TU_212" } }
      ]
    }
  ]
}
```



ПРИМЕР

Запросить события с важностью «667».

```
{"==":{"severity":667}}
```

1.10.2. Функции

1.10.2.1. MoveNext

Перемещает указатель на следующее событие в очереди.

Синтаксис

```
bool MoveNext()
```

Возвращаемое значение

«true»	Возвращается при успешном перемещении на следующий элемент очереди.
«false»	Возвращается при отсутствии следующего элемента в очереди.

Примеры



ПРИМЕР

Переместиться на следующее событие в очереди и при успешном перемещении вывести в журнал времени исполнения строку с данными текущего события.

```
moved: bool = ApAlarmQueue_1.MoveNext();
DebugTool_1.Log(moved);
if (moved)
    DebugTool_1.Log(ApAlarmQueue_1.CurrentAlarm());
```

1.10.2.2. MoveFirst

Перемещает указатель на первое событие в очереди.

Синтаксис

```
bool MoveFirst()
```

Возвращаемое значение

«true»	Возвращается при успешном перемещении на первый элемент очереди.
«false»	Возвращается при отсутствии элементов в очереди.

Примеры



ПРИМЕР

Переместиться на первое событие в очереди и при успешном перемещении вывести в журнал времени исполнения строку с данными текущего события.

```
moved: bool = ApAlarmQueue_1.MoveFirst();
DebugTool_1.Log(moved);
if (moved)
    DebugTool_1.Log(ApAlarmQueue_1.CurrentAlarm());
```

1.10.2.3. MoveNextOrFirst

Перемещает указатель на следующее событие в очереди, либо на первое, если указатель достиг конца очереди.

Синтаксис

```
bool MoveNextOrFirst()
```

Возвращаемое значение

«true»	Возвращается при успешном перемещении на следующий или первый элемент очереди.
«false»	Возвращается при отсутствии следующего элемента в очереди или пустой очереди.

Примеры



ПРИМЕР

Переместиться на следующее или первое событие в очереди и при успешном перемещении вывести в журнал времени исполнения строку с данными текущего события.

```
moved: bool = ApAlarmQueue_1.MoveNextOrFirst();
DebugTool_1.Log(moved);
if (moved)
    DebugTool_1.Log(ApAlarmQueue_1.CurrentAlarm());
```

1.10.2.4. CurrentAlarm

Получает строку с данными текущего события (на котором сейчас находится указатель).

Синтаксис

```
string CurrentAlarm()
```

Возвращаемое значение

Возвращает строку, состоящую из значений атрибутов события:

source	Источник события (тег сигнала).
condition	Имя условия генерации события.
activeTime	Время перехода состояния события в активное (в формате uint8).
sound	Звуковой файл или группа звуковых файлов, воспроизводимых при выполнении подусловия генерации сообщения о событии.



ПРИМЕР

Пример возвращаемой строки.

```
{"source":"Level.s_level", "condition":"LEVEL", "activeTime": 133032079365115751, "sound":"","sound"}
```

Примеры



ПРИМЕР

Вывести в журнал времени исполнения строку с данными текущего события.

```
DebugTool_1.Log("current: " + ApAlarmQueue_1.CurrentAlarm());
```

1.10.2.5. Reload

Запускает обновление списка активных событий с источника (объекта типа **Запрос алармов** ([стр. 86](#))).

Синтаксис

```
void Reload()
```

Примеры



ПРИМЕР

Запросить список активных событий с источника с учетом фильтра, заданного в текстовом поле.

```
ApAlarmQueue_1.JsonFilter = TextEdit_1.Text;  
ApAlarmQueue_1.Reload();
```

1.10.2.6. Clear

Очищает очередь.

Синтаксис

```
void Clear()
```


Примеры



ПРИМЕР

Очистить очередь и вывести в журнал времени исполнения длину очереди.

```
len: var = ApAlarmQueue_1.Length;  
ApAlarmQueue_1.Clear();  
DebugTool_1.Log("clear: " + String.ToString(len));
```

1.10.2.7. RemoveCurrent

Удаляет текущее событие (на котором сейчас находится указатель). При удалении события указатель сбрасывается. Для удаления другого события принудительно переведите указатель на него с помощью функций [MoveNext \(стр. 101\)](#), [MoveFirst \(стр. 102\)](#) или [MoveNextOrFirst \(стр. 102\)](#).

Синтаксис

```
void RemoveCurrent()
```

Примеры



ПРИМЕР

Удалить текущее событие и вывести в журнал времени исполнения длину очереди.

```
len: var = ApAlarmQueue_1.Length;  
ApAlarmQueue_1.RemoveCurrent();  
DebugTool_1.Log "clear: " + String.ToString(len));
```

1.10.2.8. CancelRead

Прекращает получение данных с источника (объекта типа [Запрос алармов \(стр. 86\)](#)), выполняемое функцией [Reload \(стр. 104\)](#).

Синтаксис

```
void CancelRead()
```

Примеры



ПРИМЕР

Завершить получение данных.

```
ApAlarmQueue_1.CancelRead();
```

1.10.3. События

1.10.3.1. OnActiveStateChanged

Изменение значения свойства **Активность** объекта типа **Очередь активных алармов** ([стр. 99](#)).

Параметры

activeState	bool	Состояние очереди. Соответствует значению свойства Активность .
-------------	------	------------------------------------------------------------------------

Примеры



ПРИМЕР

Выводить в журнал времени исполнения состояние очереди.

```
DebugTool_1.Log( "ACTIVE STATE CHANGED: " + String.ToString( activeState ) );
```

2. Компоненты окружения

2.1. Окружение: архиватор

2.1.1. Свойства

2.1.1.1. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

2.1.1.2. Уровень сжатия

```
UInt4 CompressionLevel
```

Степень сжатия архивируемых данных.

Значение

0	без сжатия
1	быстрый
2	нормальный
3	максимальный

2.1.1.3. Занят ли архиватор

```
bool busy
```

Позволяет проверить занят ли архиватор упаковкой или распаковкой.

Значение

«true»	архиватор занят
--------	-----------------

«false»	архиватор свободен
---------	--------------------

**ПРИМЕЧАНИЕ**

Доступно только для чтения в режиме рантайма.

2.1.2. События

2.1.2.1. PackFinished

Завершение упаковки в архив.

Параметры

error	string	текст ошибки, возникшей при упаковке
-------	--------	--------------------------------------

2.1.2.2. UnpackFinished

Завершение распаковки архива.

Параметры

error	string	текст ошибки, возникшей при распаковке
-------	--------	----------------------------------------

2.1.3. Функции

2.1.3.1. BrowseAsJSON

```
string BrowseAsJSON(string path)
```

Возвращает список файлов и папок в указанном архиве.

Параметры

path	string	путь к архиву
------	--------	---------------

Возвращаемое значение

«JSON-строка»	Строка в JSON-формате. Содержит список файлов и папок в архиве.
---------------	-----------------------------------------------------------------

Примеры



ПРИМЕР

Показать содержимое архива.

```
content: string;  
content = Archiver_1.BrowseAsJSON("C:/Tags/KP_086.7z");  
TextEdit_3 = Str.ToString(content);
```

2.1.3.2. Interrupt

```
void Interrupt()
```

Останавливает текущую операцию (упаковку файлов или распаковку архива).

Примеры



ПРИМЕР

Остановить архивирование.

```
Archiver_1.Interrupt();
```

2.1.3.3. AsyncPack

Упаковывает файлы в архив с поддержкой различных форматов, таких как 7z, zip, rar и другие.

Синтаксис

```
bool AsyncPack(string sourcePath, string destPath)
```

Параметры

sourcePath	string	путь к папке с файлами, которые должны быть упакованы в архив
destPath	string	путь к папке, в которую следует сохранить результат упаковки

Возвращаемое значение

«true»	архив успешно создан
«false»	архивирование не удалось

Примеры



ПРИМЕР

Архивировать теги КП 086.

```
result: bool;  
result = Archiver_1.AsyncPack("C:/DEMO_PROJECT/KP_086", "C:/Tags/KP_086.7z");  
TextEdit_3 = Str.ToString(result);
```

2.1.3.4. AsyncUnpack

Распаковывает указанный архив.

```
bool AsyncUnpack(string filePath, string destPath)
```

Параметры

filePath	string	путь к архиву
destPath	string	путь к папке, в которую следует распаковать архив

Возвращаемое значение

«true»	архив успешно распакован
«false»	распаковка архива не удалась

Примеры



ПРИМЕР

Распаковать архив.

```
result: bool;  
result = Archiver_1.AsyncUnpack("C:/Tags/KP_086.7z", "S:/WORK_GROUP/Tags/KP_086");  
TextEdit_3 = Str.ToString(result);
```

2.2. Окружение: группа окон

Позволяет работать с окнами в группе.

2.2.1. Свойства

2.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

2.2.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

2.2.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

2.2.1.5. Количество окон в группе

**ПРИМЕЧАНИЕ**

Доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Вывести количество окон в группе в поле ввода.

```
TextEdit.Text = Str.ToString(WindowGroup.WindowCount);
```


2.2.2. События

2.2.2.1. WindowOpened

Открытие нового окна в группе.

Параметры

windowID	string	идентификатор нового окна
----------	--------	---------------------------

2.2.2.2. WindowClosed

Закрытие окна в группе.

Параметры

windowID	string	идентификатор закрываемого окна
----------	--------	---------------------------------

2.2.3. Функции

2.2.3.1. GetWindowIDbyIndex

Определяет идентификатор окна по его индексу. Index первому элементу присваивается «0».

Параметры

Index

Примеры



ПРИМЕР

Вывести идентификатор первого окна группы в поле ввода.

```
TextEdit.Text = GetWindowIDbyIndex(0);
```

2.2.3.2. SendMessageByID

Отправляет сообщение конкретному окну, окно определяется по идентификатору.

Параметры

WindowID	int	идентификатор окна, которому будет отправлено сообщение
----------	-----	---------------------------------------------------------

MessageType	string	строковый параметр, содержащий тип сообщения. Значение должно совпадать со значением типа в событии MessageReceived (компонент Окружение: окно)
MessageData	string	строковый параметр, содержащий тело сообщения. Значение должно совпадать со значением тела в событии MessageReceived (компонент Окружение: окно)

Примеры



ПРИМЕР

Отправить сообщение на закрытие окна первому открытому окну.

```
SendMessageByID(GetWindowIDbyIndex(0), "close", "Заккрыть окно");
```

2.2.3.3. SendToAll

Отправляет сообщение всем окнам, находящимся в группе.

Параметры

MessageType	string	строковый параметр, содержащий тип сообщения. Значение должно совпадать со значением типа в событии MessageReceived (компонент Окружение: окно)
MessageData	string	строковый параметр, содержащий тело сообщения. Значение должно совпадать со значением тела в событии MessageReceived (компонент Окружение: окно)

Примеры



ПРИМЕР

Отправить сообщение на закрытие окна всем окнам.

```
SendToAll("close", "Заккрыть окно");
```

2.3. Окружение: окно

Позволяет работать с оконным окружением.

2.3.1. Свойства

2.3.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени

объекта (поля объекта) для обращения в коде.

2.3.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

2.3.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

2.3.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

2.3.1.5. Группа окон

Ссылка на компонент **Группа окон**, относительно которого будет происходить дальнейшая работа с компонентом. Указывается на вкладке **Редактор свойств**.

Идентификатор окна

```
string WindowID
```



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести идентификатор окна в поле ввода.

```
TextEdit.Text = Window.WindowID;
```

2.3.2. События

2.3.2.1. MessageReceived

Получение сообщения.

Параметры

messageType	string	тип полученного сообщения (определяется пользователем). Значение параметра должно совпадать с типом сообщения в функции, отправляющей сообщение окну
messageData	string	тело сообщения (определяется пользователем). Значение параметра может использоваться для коммуникаций между окнами

2.4. Окружение: сеть

Позволяет работать с сетевым окружением.

2.4.1. Свойства

2.4.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.4.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

2.4.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

2.4.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

2.4.1.5. Полное имя компьютера

```
string FullComputerName
```

Полное имя сетевого компьютера в домене.

**ПРИМЕЧАНИЕ**

Доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Вывести имя сетевого компьютера в поле ввода.

```
TextEdit.Text = Network.FullComputerName;
```

2.4.1.6. Имя компьютера

```
string ComputerName
```

Имя сетевого компьютера.

**ПРИМЕЧАНИЕ**

Доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Вывести имя сетевого компьютера в поле ввода.

```
TextEdit.Text = Network.ComputerName;
```

2.5. Окружение: файлы

Компонент позволяет работать с каталогами файловой системы, читать и записывать содержимое текстовых файлов.

2.5.1. Свойства

2.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	Одиночный объект.
>«1»	Массив соответствующей размерности.

2.5.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```


2.5.1.4. Индекс

```
int8 Index
```

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

2.5.1.5. Папка текущего пользователя

```
string CurrentUserFolder
```

Полный путь к папке текущего пользователя операционной системы.

Примеры



ПРИМЕР

Вывести полный путь к папке текущего пользователя ОС в текстовое поле.

```
Text_1 = FileSystem.CurrentUserFolder;
```

2.5.1.6. Общая папка пользователей

```
string AllUsersFolder
```



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Полный путь к общей папке пользователей операционной системы.

Примеры



ПРИМЕР

Вывести полный путь к общей папке пользователей ОС в текстовое поле.

```
Text_1 = FileSystem.AllUsersFolder;
```



ПРИМЕЧАНИЕ

В ОС Linux общая папка пользователей отсутствует, поэтому в данных системах свойство возвращает путь к папке текущего пользователя.

2.5.1.7. Папка текущего проекта

Полный путь к папке проекта SePlatform.HMI.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести полный путь к проекту в текстовое поле.

```
Text_1 = FileSystem.ProjectFolder;
```

2.5.1.8. Ошибка

```
FileSystemError Error
```

При выполнении операций с файловой системой возможно возникновение ошибок. Чтобы компонент **Окружение: файлы** ([стр. 120](#)) мог передавать элементу **Окружение: ошибка работы с файлами** ([стр. 143](#)) сообщения об ошибках, внутри свойства **Ошибка** указывается ссылка на компонент **Окружение: ошибка работы с файлами** ([стр. 143](#)).

Чтобы узнать детали возникшей ошибки, обратитесь к свойствам **Окружение: ошибка работы с файлами** ([стр. 143](#)):

- Состояние ([стр. 143](#)) - произошла ли ошибка при возникновении последней операции;
- Текст ([стр. 144](#)) - текст ошибки последней операции;
- Код ([стр. 145](#)) - код ошибки последней операции;



ПРИМЕР

Вывести возникшую ошибку в поле ввода.

```
LineEdit_1.Text = FileSystemError_1.Text;
```

2.5.1.9. Корневой путь

```
string RootPath
```

Корневой путь определяет базовую директорию, в пределах которой объект может выполнять операции с файловой системой. Этот путь служит ориентиром для работы с файлами и папками. При попытке совершить файловую операцию за пределами указанного пути в свойстве **Корневой путь** ([стр. 123](#)) возникнет ошибка, которую можно увидеть при обращении к свойствам **Окружение: ошибка работы с файлами** ([стр. 143](#)).

2.5.2. Функции

2.5.2.1. Exists

Определяет существование файла или папки.

Синтаксис

```
bool Exists(string Path)
```


Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

«true»	Возвращается в случае, если файл (или папка) с заданным в параметре Path существует.
«false»	Возвращается в случае, если файл (или папка) с заданным в параметре Path не существует.

Примеры

 **ПРИМЕР**
Если по указанному пути файл существует, вывести в поле ввода: «Файл существует».

```
if (FileSystem.Exists("C:\\Work\\Test\\log.txt"))  
{  
    TextEdit.Text = "Файл существует";  
};
```

2.5.2.2. CombinePath

Объединяет указанные текстовые пути в единый путь. Позволяет облегчить формирование составных путей. Первый параметр `left` представляет основной путь, а второй параметр `right` представляет дополнительный путь до папки или текстового файла.

Все пути указываются в двойных кавычках. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Синтаксис

```
string CombinePath(string left, string right)
```

Параметры

Параметр	Тип	Описание
left	string	Текстовый путь 1.

right	string	Текстовый путь 2.
-------	--------	-------------------

Возвращаемое значение

Возвращает строку, представляющую объединенный путь.

Примеры



ПРИМЕР

Объединить указанные пути и записать в поле ввода. Результат слияния - строка: «C:\Users\пользователь\TestFolder\TestFile.txt».

```
TextEdit.Text = FileSystem.CombinePath("C:\\Users\\пользователь",  
    "\\TestFolder\\TestFile.txt");
```

2.5.2.3. CreateRecursiveFolder

Создает папку по указанному пути.

Синтаксис

```
void CreateRecursiveFolder(string folderPath)
```

Параметры

Параметр	Тип	Описание
folderPath	string	Путь к создаваемой папке. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Примеры



ПРИМЕР

Создать папку.

```
FileSystem.CreateRecursiveFolder("C:\\Work\\Test\\");
```

2.5.2.4. RemoveRecursive

Удаляет папку, включая все файлы и подпапки внутри неё. Используется также для удаления отдельного файла по указанному пути.

Синтаксис

```
void RemoveRecursive(string sourcePath)
```

Параметры

Параметр	Тип	Описание
sourcePath	string	Путь до папки или файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Примеры



ПРИМЕР

Удалить папку Work по указанному пути, включая все вложенные папки и файлы.

```
FileSystem.RemoveRecursive("C:\\Work\\");
```

Удалить файл log.txt по указанному пути.

```
FileSystem.RemoveRecursive("C:\\Work\\log.txt");
```

2.5.2.5. ReadTextFile

Предоставляет содержимое текстового файла для чтения. Содержимое файла должно быть в кодировке «UTF-8».

Синтаксис

```
string ReadTextFile(string FilePath)
```

Параметры

Параметр	Тип	Описание
----------	-----	----------

FilePath	string	Путь до текстового файла. При передаче пути необходимо учесть, что обратные слэши должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слэши без экранирования (например, C:/project/file.txt).
----------	--------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Возвращаемое значение

Возвращает строку с содержимым текстового файла.

Примеры



ПРИМЕР

Содержимое файла поместить в текстовое поле. В параметре FilePath указать путь до текстового файла.

```
TextEdit.Text = FileSystem.ReadTextFile("C:\\Work\\Test\\log.txt");
```

2.5.2.6. AppendTextFile

Записывает строку текста в конец текстового файла (*.txt, *.xml, *.json и другие любые текстовые форматы).

Синтаксис

```
void AppendTextFile(string filePath, string content)
```

Параметры

Параметр	Тип	Описание
FilePath	string	Путь до текстового файла. При передаче пути необходимо учесть, что обратные слэши должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слэши без экранирования (например, C:/project/file.txt).
Content	string	Текстовая строка.

Примеры



ПРИМЕР

Записать строку текста в файл по указанному пути.

```
FileSystem_1.AppendTextFile("C:\\Work\\Test\\log.txt", "Строка, которая будет  
записана в конец файла");
```

2.5.2.7. ReadBinaryAsJSON

Предоставляет содержимое бинарного файла для чтения.

Синтаксис

```
string ReadBinaryAsJSON(string FilePath, string Type, uint8 Offset, uint2 Count)
```

Параметры

Параметр	Тип	Описание
FilePath	string	Путь к бинарному файлу. При передаче пути необходимо учесть, что обратные слэши должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слэши без экранирования (например, C:/project/file.txt).
Type	string	Тип элементов бинарного файла.
Offset	uint8	Позиция, с которой нужно начать чтение.
Count	uint2	Длина отрезка данных.

Возвращаемое значение

Возвращает строку в формате JSON с содержимым из бинарного файла.

Примеры



ПРИМЕР

Считать из бинарного файла «bin.dat» в текстовое поле 20 элементов типа «int2». Считывание начинать с 100 позиции.

```
TextEdit.Text = FileSystem.ReadBinaryAsJSON("bin.dat", "int2", 100, 20);
```


2.5.2.8. WriteTextFile

Создает текстовый файл *.txt и записывает в него указанную строку.

Синтаксис

```
void WriteTextFile(string FilePath, string Content)
```

Параметры

Параметр	Тип	Описание
FilePath	string	Путь к текстовому файлу. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).
Content	string	Текстовая строка.

Примеры



ПРИМЕР

Записать строку текста в файл по указанному пути.

```
Files.WriteTextFile("C:\\Work\\Test\\log.txt", "Содержимое для текстового файла");
```

2.5.2.9. CanonicalPath

Нормализует переданный путь (приводит разделитель пути к «/» и убирает переходы вверх «..»).

Синтаксис

```
string CanonicalPath(string path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя.

Возвращаемое значение

Возвращает строку, содержащую нормализованный путь.

Примеры



ПРИМЕР

Привести указанный путь к нормализованному виду. Результат поместить в текстовое поле.

```
TextEdit.Text = FileSystem_1.CanonicalPath("C:\\Work\\..");  
// Результат выполнения: "C:/"
```

2.5.2.10. ToGeneric

Заменяет все разделители пути на единый «/».

Синтаксис

```
string ToGeneric(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя.

Возвращаемое значение

Возвращает строку с путем, в котором все разделители заменены на «/».

Примеры



ПРИМЕР

Вернуть в текстовое поле строку пути с заменой разделителей на единый «/».

```
TextEdit.Text = FileSystem_1.ToGeneric("C:\\Work\\Test\\..");  
// Результат выполнения: "C:/Work/Test/.."
```

2.5.2.11. ToNative

Заменяет единый «/» разделитель частей пути на платформенный.

Синтаксис

```
string ToNative(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя.

Возвращаемое значение

Возвращает строку с путем, в котором единые разделители заменены на платформенный.

Примеры



ПРИМЕР

Вернуть в текстовое поле строку пути с заменой разделителей на используемые ОС.

```
TextEdit.Text = FileSystem_1.ToNative("C:/Work/Test/..");  
// Результат выполнения на ОС Windows: "C:\Work\Test\.."  
// Результат выполнения на ОС Linux: "C:/Work/Test/.."
```

2.5.2.12. PreferredSeparator

Возвращает используемый ОС разделитель в строке пути.

Синтаксис

```
string PreferredSeparator()
```

Возвращаемое значение

Возвращает строку с символом разделителя, используемого операционной системой для путей.

Примеры



ПРИМЕР

Вернуть в текстовое поле используемый ОС разделитель элементов в строке пути.

```
TextEdit.Text = FileSystem_1.PreferredSeparator();  
// Результат выполнения на ОС Windows: "\"  
// Результат выполнения на ОС Linux: "/"
```

2.5.2.13. ParentPath

Возвращает родительский путь в пределах корневого пути, заданного в свойстве [Корневой путь](#) (стр. 123).

Синтаксис

```
string ParentPath(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

Возвращает строку с родительским путем в пределах корневого пути. В случае отсутствия родительского пути возвращает точку «.»

Примеры



ПРИМЕР

Вернуть в текстовое поле родительский путь.

```
TextEdit.Text = FileSystem_1.ParentPath("C:\\Work\\Test\\");  
// Результат выполнения: "C:/"
```

2.5.2.14. HasFileName

Проверяет, есть ли имя файла по указанному пути.

Синтаксис

```
bool HasFileName(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

«true»	В указанном пути содержится имя файла (путь указывает на файл).
«false»	В указанном пути не содержится имени файла (путь указывает на директорию).

Примеры



ПРИМЕР

Если для указанного пути имя файла существует, вывести в поле ввода: «Имя файла существует».

```
if (FileSystem_1.HasFileName("C:\\Work\\Test\\file.txt"))
{ TextEdit.Text = "Имя файла существует";
};
```

2.5.2.15. HasParentPath

Есть ли родительский путь в пределах корневого пути, заданного в свойстве [Корневой путь \(стр. 123\)](#).

Синтаксис

```
bool HasParentPath(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя. При передаче пути необходимо учесть, что обратные слэши должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слэши без экранирования (например, C:/project/file.txt).

Возвращаемое значение

«true»	Родительский путь существует.
«false»	Родительского пути не существует.

Примеры

ПРИМЕР

Если для указанного пути родитель существует, вывести в поле ввода: «Родитель существует».

```
if (FileSystem_1.HasParentPath("C:\\Work\\Test\\.."))
{
    TextEdit.Text = "Родитель существует";
};
```

2.5.2.16. IsParent

Является ли parentPath родителем childPath в пределах корневого пути, заданного в свойстве [Корневой путь](#) [\(стр. 123\)](#).
При передаче путей для параметров parentPath и childPath необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Синтаксис

```
bool IsParent(string parentPath, string childPath)
```

Возвращаемое значение

«true»	parentPath является родителем для childPath.
«false»	parentPath не является родителем для childPath.

Параметры

Параметр	Тип	Описание
parentPath	string	Родительский путь до папки или файла пользователя.
childPath	string	Дочерний путь до папки или файла пользователя.

Примеры

ПРИМЕР

Проверить, является ли первый путь родителем второго. Результат вывести в журнал исполнения.

```
DebugTool_1.Log(String.ToString(FileSystem_1.IsParent("C:\\Work\\" ,
"C:\\Work\\Test\\..")));
// Результат выполнения: "false"
```

2.5.2.17. AbsolutePath

Возвращает полный абсолютный путь (достроенный с помощью [Корневой путь \(стр. 123\)](#)) для переданного относительного.

При корневом пути по умолчанию «/» принимаются все полные пути.

При иных значениях [Корневой путь \(стр. 123\)](#) принимаются только относительные пути.

Синтаксис

```
string AbsolutePath(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя.

Возвращаемое значение

Возвращает строку с полным абсолютным путем, сконструированным с использованием значения свойства [Корневой путь \(стр. 123\)](#) для переданного относительного пути.

Примеры



ПРИМЕР

Вернуть в текстовое поле абсолютный путь для переданного относительного.

```
// RootPath = "C:/"
TextEdit.Text = FileSystem_1.AbsolutePath("Work/Test");
// Результат выполнения: "C:/Work/Test"
```

2.5.2.18. RelativePath

Возвращает относительный путь (часть без [Корневой путь \(стр. 123\)](#)) для переданного абсолютного.

Синтаксис

```
string RelativePath(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

Возвращает строку с относительным путем, представляющим часть пути без учета значения свойства **Корневой путь** (стр. 123), для переданного абсолютного пути.

Примеры



ПРИМЕР

Вернуть в текстовое поле относительный путь для переданного абсолютного.

```
TextEdit.Text = FileSystem_1.RelativePath("C:\\Work\\Test\\");  
// Результат выполнения: "C:/Work/Test"
```

2.5.2.19. FileName

Предоставляет название объекта файловой системы, переданного по пути.

Синтаксис

```
string FileName(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

Возвращает строку с названием файла из переданного пути. Если путь не содержит файла (только папки), возвращается пустая строка.

Примеры



ПРИМЕР

Вернуть в текстовое поле полное имя файла из переданного пути.

```
TextEdit.Text = FileSystem_1.FileName("C:\\Work\\Test\\v8.dll");  
// Результат выполнения: "v8.dll"
```

2.5.2.20. Extension

Возвращает расширение элемента файловой системы.

Синтаксис

```
string Extension(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

Возвращает строку, содержащую расширение файла из переданного пути. Если путь не содержит расширения (например, путь указывает на папку), то возвращается пустая строка.

Примеры



ПРИМЕР

Вернуть в текстовое поле только расширение файла из переданного пути

```
TextEdit.Text = FileSystem_1.Extension("C:\\Work\\Test\\v8.dll");  
// Результат выполнения: ".dll"
```

2.5.2.21. Stem

Возвращает имя файла без расширения.

Синтаксис

```
string Stem(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

Возвращает строку, содержащую название файла без расширения из переданного пути.

Примеры



ПРИМЕР

Вернуть в текстовое поле только имя файла без расширения.

```
TextEdit.Text = FileSystem_1.Stem("C:\\Work\\Test\\v8.dll");  
// Результат выполнения: "v8"
```

2.5.2.22. IsFile

Указывает ли путь на файл.

Синтаксис

```
bool IsFile(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

«true»	Возвращается в случае, если файл с заданным в параметре Path существует.
«false»	Возвращается в случае, если файл с заданным в параметре Path не существует.

Примеры



ПРИМЕР

Проверить, указывает ли переданный путь на файл. Результат вывести в журнал исполнения.

```
DebugTool_1.Log(String.ToString(FileSystem_1.IsFile("C:\\Work\\Test\\v8.dll")));  
// Результат выполнения: true
```

2.5.2.23. IsDirectory

Указывает ли путь на директорию.

Синтаксис

```
bool IsDirectory(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

«true»	Возвращается в случае, если директория с заданным в параметре Path существует.
«false»	Возвращается в случае, если директория с заданным в параметре Path не существует.

Примеры



ПРИМЕР

Проверить, указывает ли переданный путь на директорию. Результат вывести в журнал исполнения.

```
DebugTool_1.Log(String.ToString(FileSystem_1.IsDirectory("C:\\Work\\Test\\")));  
// Результат выполнения: true
```

2.5.2.24. IsSymLink

Указывает ли путь на символическую ссылку (симлинк).

Синтаксис

```
bool IsSymLink(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до папки или файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

«true»	Возвращается в случае, если симлинк с заданным в параметре Path существует.
«false»	Возвращается в случае, если симлинк с заданным в параметре Path не существует.

Примеры



ПРИМЕР

Проверить, указывает ли переданный путь на символическую ссылку. Результат вывести в журнал исполнения.

```
DebugTool_1.Log(String.ToString(FileSystem_1.IsSymLink("C:\\Work\\Test\\")));  
// Результат выполнения: false
```

2.5.2.25. FileSize

Предоставляет сведения о размере файла.

Синтаксис

```
uint8 FileSize(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до файла пользователя. При передаче пути необходимо учесть, что обратные слешы должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слешы без экранирования (например, C:/project/file.txt).

Возвращаемое значение

Возвращает значение типа uint8, представляющее размер файла в байтах. Если путь указывает на директорию или ссылку, возвращается «0».

Примеры



ПРИМЕР

Вывести в журнал исполнения размер файла в байтах.

```
DebugTool_1.Log(String.ToString(FileSystem_1.FileSize("C:\\Work\\Test\\v8.dll")));  
// Результат выполнения: 20 914 176
```

2.5.2.26. CreatedTime

Предоставляет сведения о времени создания элемента файловой системы.

Для возврата даты-времени в виде строки необходимо использовать функции для работы с метками времени. Подробнее об этом описано в документации на SePlatform.Om.

Синтаксис

```
timestamp CreatedTime(string Path)
```

Параметры

Параметр	Тип	Описание
----------	-----	----------

Path	string	Путь до файла пользователя. При передаче пути необходимо учесть, что обратные слэши должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слэши без экранирования (например, C:/project/file.txt).
------	--------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Возвращаемое значение

Возвращает значение типа timestamp, представляющее время создания элемента файловой системы.

Примеры

ПРИМЕР

Вывести в текстовое поле время создания файла.

```
TextEdit.Text = DateTime.ToString(FileSystem_1.CreatedTime("C:\\Work\\Test\\v8.dll"), "%d/%m/%Y-%H:%M:%S.%f");  
// Результат выполнения: "12/03/2023-23:42:52.000000";
```

2.5.2.27. ModifiedTime

Предоставляет сведения о времени последнего изменения элемента файловой системы.

Для возврата даты-времени в виде строки необходимо использовать функции для работы с метками времени. Подробнее об этом описано в документации на SePlatform.Om.

Синтаксис

```
timestamp ModifiedTime(string Path)
```

Параметры

Параметр	Тип	Описание
Path	string	Путь до файла пользователя. При передаче пути необходимо учесть, что обратные слэши должны быть экранированы (например, C:\\project\\file.txt). Также можно использовать прямые слэши без экранирования (например, C:/project/file.txt).

Возвращаемое значение

Возвращает значение типа timestamp, представляющее время последнего изменения элемента файловой системы.

Примеры



ПРИМЕР

Вывести в текстовое поле время последнего изменения файла.

```
TextEdit.Text = DateTime.ToString(FileSystem_1.ModifiedTime  
("C:\\Work\\Test\\v8.dll"), "%d/%m/%Y-%H:%M:%S.%f");  
// Результат выполнения: "12/03/2023-23:42:52.000000";
```

2.6. Окружение: ошибка работы с файлами

Компонент позволяет хранить информацию об ошибке, возникшей в результате выполнения какой-либо операции с файлом.

Объекты типа **Окружение: файлы** ([стр. 120](#)) и **Окружение: браузер файлов** ([стр. 149](#)) могут использовать свойства компонента **Окружение: ошибка работы с файлами** ([стр. 143](#)) для взаимодействия с данной ошибкой.

2.6.1. События

2.6.1.1. Occured

Возникновение ошибки в компонентах **Окружение: файлы** ([стр. 120](#)) и **Окружение: браузер файлов** ([стр. 149](#)).

Параметры

errorCode	unit4	Код ошибки последней операции.
errorText	string	Текст ошибки последней операции.

2.6.2. Свойства

2.6.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.6.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	Одиночный объект.
>«1»	Массив соответствующей размерности.

2.6.2.3. Состояние

bool State



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Произошла ли ошибка при выполнении последней операции.

Значение

«true»	При выполнении последней операции возникла ошибка.
«false»	При выполнении последней операции ошибки не возникло.

Примеры



ПРИМЕР

Проверить были ли ошибки при выполнении последней операции. Результат вывести в журнал исполнения.

```
DebugTool_1.Log(FileSystem_1.Error.State);
```

2.6.2.4. Текст

String Text

Текст ошибки последней операции.

В случае отсутствия ошибок будет выведена пустая строка " ".



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в журнал исполнения текст ошибки при выполнении последней операции.

```
DebugTool_1.Log(FileSystem_1.Error.Text);
```

2.6.2.5. Код

uint4 Code

Код ошибки последней операции.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в журнал исполнения код ошибки последней операции

```
DebugTool_1.Log(FileSystem_1.Error.Code);
```

2.7. Окружение: система

Компонент позволяет получить информацию о текущей операционной системе, на которой работает приложение SePlatform.HMI.

2.7.1. Свойства

2.7.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.7.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиначный объект
>«1»	массив соответствующей размерности

2.7.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

2.7.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

2.7.1.5. Название

```
string Name
```

Название текущей операционной системы.

Примеры

**ПРИМЕР**

Вывести информацию о названии операционной системы в текстовое поле

```
Text_1.Text = System_1.Name;
```

2.7.1.6. Семейство

```
string Family
```

Семейство текущей операционной системы.

Примеры

**ПРИМЕР**

Вывести информацию о семействе операционной системы в текстовое поле

```
Text_1.Text = System_1.Family;
```

2.7.1.7. Версия

```
string Version
```

Номер версии текущей операционной системы.

Примеры



ПРИМЕР

Вывести информацию о версии операционной системы в текстовое поле.

```
Text_1.Text = System_1.Version;
```

2.7.1.8. Ядро

```
string Kernel
```

Информация о ядре текущей операционной системы.

Примеры



ПРИМЕР

Вывести информацию о ядре операционной системы в текстовое поле.

```
Text_1.Text = System_1.Kernel;
```

2.7.1.9. Версия ядра

```
string KernelVersion
```

Версия ядра текущей операционной системы.

Примеры



ПРИМЕР

Вывести информацию о версии ядра операционной системы в текстовое поле.

```
Text_1.Text = System_1.KernelVersion;
```

2.7.1.10. Порядок байтов

```
uint1 Endianness
```

Порядок следования байтов в памяти операционной системы.

Значение

«0»	«BigEndian» - старшие байты хранятся в начале адреса памяти, а младшие - в конце.
«1»	«LittleEndian» - младшие байты хранятся в начале адреса памяти, а старшие - в конце.

Примеры



ПРИМЕР

Вывести информацию о порядке байтов в текстовое поле.

```
Text_1.Text = (System_1.Endianness == 0 ? "BigEndian" : "LittleEndian");
```

2.8. Окружение: браузер файлов

Компонент позволяет просматривать содержимое файловой системы.

2.8.1. Функции

2.8.1.1. Browse

Начинает новую сессию просмотра файловой системы для переданного относительного пути.

Синтаксис

```
void Browse(string path)
```

Параметры

path	Путь, с которого начинается браузеринг.
------	-----------------------------------------



ПРИМЕЧАНИЕ

Для отображения результатов браузеринга подключите **Окружение:браузер файлов** ([стр. 149](#)) к одному из компонентов: **Дерево** или **Таблица**.

2.8.2. Свойства

2.8.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.8.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	Одиночный объект.
>«1»	Массив соответствующей размерности.

2.8.2.3. Файлы

FileSystem FileSystem

Используется для считывания значения **Корневой путь** ([стр. 123](#)) при передаче объекта типа **Окружение : файлы** ([стр. 120](#))



ПРИМЕР

Вывести в журнал исполнения значение **Корневой путь** ([стр. 123](#)) для объекта **Окружение : файлы** ([стр. 120](#)).

```
DebugTool_1.Log(FileSystemBrowser_1.FileSystem.RootPath);
```

2.8.2.4. Ошибка

FileSystemError Error

Используется для хранения информации об ошибке, возникшей в результате выполнения какой-либо операции при просмотре файловой системы. Для просмотра возникшей ошибки, в редакторе свойств указывается ссылка на компонент **Окружение : ошибка работы с файлами** ([стр. 143](#)). Чтобы узнать детали возникшей ошибки, обратитесь к свойствам **Окружение : ошибка работы с файлами** ([стр. 143](#)).

Примеры



ПРИМЕР

Вывести в журнал исполнения текст ошибки при выполнении последней операции.

```
DebugTool_1.Log(FileSystem_1.Error.Text);
```

2.8.2.5. Рекурсивный браузинг

```
bool Recursive
```

Позволяет выполнять операции браузинга рекурсивно (операция браузинга будет выполняться не только в указанной директории, но и в ее поддиректориях, и так далее по всем вложенным уровням директорий).

Значение

«true»	Прогружает все дерево заданного пути.
«false»	Прогружает только верхний уровень папок.

2.8.2.6. Показывать в имени файлов их расширения

```
bool WithExtension
```

Отображать ли имя файла вместе с расширением.

Значение

«true»	Отображение только имени файла («text.»).
«false»	Отображение имени файла вместе с расширением («text.txt»).

2.8.2.7. Отображать в браузере папки

```
bool IncludeFolders
```

Отображать ли папки при браузинге.

Значение

«true»	Отображение папок включено.
«false»	Отображение папок выключено.

2.8.2.8. Отображать в браузере файлы

Отображать ли в файлы при браузинге

Значение

«true»	Отображение файлов включено.
«false»	Отображение файлов выключено.

2.8.2.9. Отображать в браузере ссылки

```
bool IncludeSymLinks
```

Отображать ли символические ссылки при браузинге файловой системы.

Символические ссылки позволяют создавать ярлыки на файлы и директории и сослаться на них из других мест в файловой системе.

Значение

«true»	Отображение символических ссылок включено.
«false»	Отображение символических ссылок выключено.

2.8.2.10. Показывать скрытые элементы

```
bool ShowHiddenObjects
```

Позволяет просматривать скрытые элементы при браузинге.

Значение

«true»	Отображение скрытых элементов включено.
«false»	Отображение скрытых элементов выключено.

2.8.2.11. Показывать системные элементы

```
bool ShowSystemObjects
```

Позволяет включить отображение системных элементов при браузинге файловой системы.

К системным элементам относятся файлы и папки, которые обычно скрыты в файловой системе, поскольку содержат в себе данные, не предназначенные для прямого доступа пользователей. Например, в ОС Windows системные элементы могут включать папки /ProgramData и /AppData, а в Linux - папки /bin, /sbin и т.д.

Значение

«true»	Отображение системных элементов включено.
«false»	Отображение системных элементов выключено.

2.8.2.12. Фильтр на отображаемые объекты ФС

Позволяет включить фильтр на отображение файлов, папок и ссылок вместе с расширением.



ПРИМЕР

Отфильтровать файлы только с «.txt» расширением

```
*.txt
```

Отфильтровать файлы, где в имени есть «qwer»

```
*qwer*
```

2.9. Окружение: процесс

2.9.1. События

2.9.1.1. StateChanged

Изменение состояния процесса, запущенного с помощью **Окружение: процесс**.

Параметры

state	UInt1	Состояние процесса.
-------	-------	---------------------

Значение

«0»	Процесс выключен
«1»	Процесс запускается.
«2»	Процесс запущен.



ПРИМЕР

Используя параметр «state» события **StateChanged** вывести в текстовое поле сообщение о начале работы процесса.

```
if (state == 2) {  
    Text_1.Text = "the process is running";  
}
```

2.9.1.2. Started

Процесс был запущен и начал свою работу.

Параметры

id	UInt4	Идентификатор запущенного процесса.
----	-------	-------------------------------------

2.9.1.3. Finished

Процесс завершил свою работу.

Параметры

exitCode	UInt4	Код завершившегося процесса.
----------	-------	------------------------------

Значение

«0»	Процесс успешно завершился.
«>0»	Процесс завершился с ошибкой.

2.9.1.4. StartFailed

Процесс не запущен из-за возникшей ошибки.

Параметры

error	string	Описание ошибки, которая привела к неудачному запуску процесса.
-------	--------	-----------------------------------------------------------------

2.9.1.5. OnStdout

В стандартный поток вывода «**stdout**» поступили новые данные от процесса, запущенного с помощью
Окружение: процесс.

Параметры

value	string	Данные из стандартного потока вывода « stdout » процесса.
-------	--------	------------------------------------------------------------------

2.9.1.6. OnStderr

В стандартный поток вывода ошибок «**stderr**» поступили новые данные, от процесса, запущенного с помощью
Окружение: процесс.

Параметры

value	string	Данные из стандартного потока вывода ошибок « stderr ».
-------	--------	----------------------------------------------------------------

2.9.2. Функции

2.9.2.1. Run

Запускает процесс с использованием текущих настроек компонента **Окружение: процесс**. При вызове этой функции процесс начинает свою работу, выполняя указанное исполняемое приложение с переданными аргументами командной строки и в указанном рабочем каталоге.

Функцию **Run()** следует вызывать после того, как вы настроили необходимые параметры процесса, такие как [Путь к исполняемому файлу](#) и [Аргументы запуска](#).

Синтаксис

```
void Run();
```

Примеры



ПРИМЕР

Запустить «**example.txt**» с помощью приложения «**notepad.exe**» из системной папки `C:\Windows\System32\`. Окно запущенного приложения сделать видимым.

```
Process_1.Path = "C:\\Windows\\System32\\notepad.exe";  
Process_1.Arguments = "C:\\Work\\example.txt";  
Process_1.ShowWindow = true;  
Process_1.Run();
```

2.9.2.2. Terminate

Завершает процесс, который был запущен с помощью компонента **Окружение : процесс**. При вызове этой функции процесс прекращает свою работу, и его выполнение завершается.

Синтаксис

```
void Terminate();
```

2.9.2.3. Write

Используется для передачи данных в запущенный процесс и взаимодействия с ним через ввод данных.

Синтаксис

```
void Write(string message);
```

Параметры

message	string	Строка с данными, которая будет передана в запущенный процесс.
---------	--------	----------------------------------------------------------------

Примеры



ПРИМЕР

Запустить исполняемый файл 1.bat, который ожидает ввода имени от пользователя. Передать в качестве аргумента имя «UserName».

```
Process_1.Path = "C:\\Work\\1.bat";  
Process_1.ShowWindow = true;  
Process_1.Run();
```

В файле 1.bat содержится следующий код:

```
@echo off  
set /p name=Введите ваше имя:  
echo Привет, %name%! Добро пожаловать!  
pause
```

Используем функцию **Write()** для передачи имени «UserName» в режиме рантайма:

```
Process_1.Write("UserName");
```

После выполнения данного примера, в окне командной строки вы увидите:

```
Введите ваше имя:  
Привет, UserName! Добро пожаловать!
```

2.9.2.4. ClearAccumulatedStdout

Используется для очистки накопленного стандартного вывода «stdout». После вызова этой функции свойство **Накопленный стандартный вывод** будет пустым.

Синтаксис

```
void ClearAccumulatedStdout();
```

2.9.2.5. ClearAccumulatedStderr

Используется для очистки накопленного стандартного вывода ошибок «stderr». После вызова этой функции свойство **Накопленный стандартный вывод ошибок** будет пустым.

Синтаксис

```
void ClearAccumulatedStderr();
```

2.9.3. Свойства

2.9.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.9.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

2.9.3.3. Переменные среды процесса

Ссылка на компонент **Окружение: переменные среды процесса**.

Указание ссылки дает возможность использовать настройки переменных окружения, определенных в компоненте **Окружение: переменные среды процесса** при запуске процесса с помощью компонента **Окружение: процесс**. Если ссылка на **Окружение: переменные среды процесса** указана, но переменные окружения не настроены, то будут использоваться значения переменных окружения по умолчанию из операционной системы.

Примеры



ПРИМЕР

Записать в поле ввода значение переменной среды «TEMP»

```
TextEdit.Text = Process_1.Environment.Get("TEMP");
```

2.9.3.4. ID

Доступно только для чтения в режиме рантайма.

```
UInt4 ID
```

ID процесса SePlatform.HMI.

Примеры



ПРИМЕР

Вывести ID процесса SePlatform.HMI в поле ввода.

```
TextEdit.Text = String.ToString(Process_1.ID);
```

2.9.3.5. State

Uint1 State

Позволяет отслеживать состояние процесса, запущенного с помощью компонента **Окружение: процесс**.

Значение

«0»	Процесс выключен
«1»	Процесс запускается.
«2»	Процесс запущен.

2.9.3.6. ExitCode

int4 ExitCode

Код завершения процесса после его выполнения.

Значение

«0»	Процесс успешно завершился.
«>0»	Процесс завершился с ошибкой.

2.9.3.7. Путь к исполняемому файлу

Доступно только для чтения в режиме рантайма.

string ExecutablePath

Предоставляет полный путь к исполняемому файлу.

Примеры

ПРИМЕР

Вывести в текстовое поле путь к исполняемому файлу.

```
Text_2.Text = Process_1.ExecutablePath;
```

2.9.3.8. Путь до файла приложения

```
string Path
```

Позволяет указать путь к исполняемому файлу процесса, который будет запускаться с помощью компонента **Окружение: процесс**. Установка значения этого свойства определяет, какое приложение должно быть запущено при вызове метода **Run()**.

Примеры

ПРИМЕР

Установить путь к исполняемому файлу «MyApp.exe»

```
Process_2.Path = "C:\\Program Files\\MyApp\\MyApp.exe";
```

2.9.3.9. Показывать окно

```
bool ShowWindow
```

Отображать ли окно запускаемого внешнего процесса при его выполнении.

Значение

«true»	Окно запускаемого процесса будет отображаться на экране.
«false»	Запускаемый процесс будет выполняться в фоновом режиме, без отображения его окна на экране

Примеры



ПРИМЕР

Запустить приложение «MyApp.exe» с видимым окном.

```
Process_2.ShowWindow = true;  
Process_2.Path = "C:\\Program Files\\MyApp\\MyApp.exe";  
Process_2.Run();
```

2.9.3.10. Аргументы запуска

```
string Arguments
```

Позволяет указать дополнительные аргументы командной строки, которые будут переданы приложению при его запуске с помощью компонента **Окружение: процесс**. Аргументы командной строки представляют собой дополнительные параметры, которые влияют на работу запускаемого процесса.



ОБРАТИТЕ ВНИМАНИЕ

Если аргументы содержат пробелы, необходимо заключить их в двойные кавычки. Это важно для корректной интерпретации параметров процессом. Например, если вам нужно передать аргумент D:\\Project Test\\hello.py содержащий пробел, то свойство **Аргументы запуска** следует установить так:

```
Process_1.Arguments = "\"D:\\Project Test\\hello.py\"";
```

Примеры



ПРИМЕР

Открыть текстовый файл «1.txt» с помощью «notepad.exe»

```
Process_1.Path = "C:\\Windows\\System32\\notepad.exe";  
Process_1.Arguments = "C:\\Work\\1.txt";  
Process_1.ShowWindow = true;  
Process_1.Run();
```

2.9.3.11. Рабочий каталог

```
string WorkingDirectory
```

Позволяет задать рабочий каталог для запускаемого внешнего процесса.

Рабочий каталог представляет собой директорию, в которой будет выполняться запускаемый процесс.

Используется, если запускаемому процессу необходим доступ к файлам или ресурсам, расположенным в определенной директории, и его работа зависит от этой локации.

Примеры

ПРИМЕР

Задать рабочий каталог «C:\Program Files\Work\Process» для запускаемого процесса.

```
Process_1.WorkingDirectory = "C:\\Program Files\\Work\\Process";
```

2.9.3.12. Кодировка

String Encoding

Позволяет указать кодировку, которая будет использоваться при выполнении внешнего процесса. Кодировка определяет, как символы текста будут представлены в байтовом виде.

Примеры

ПРИМЕР

Открыть текстовый файл 1.txt с помощью notepad.exe с содержимым в кодировке «UTF-8»

```
Process_1.Path = "C:\\Windows\\System32\\notepad.exe";
Process_1.Arguments = "C:\\Work\\1.txt";
Process_1.ShowWindow = true;
Process_1.Encoding = "UTF-8"
Process_1.Run();
```

2.9.3.13. Накапливать стандартный вывод

bool AccumulateStdout

Указывает, следует ли накапливать стандартный вывод «stdout» при выполнении процесса, запущенного с использованием компонента **Окружение: процесс**. Когда данное свойство установлено в значение «true», компонент **Окружение: процесс** будет накапливать данные, выводимые в стандартный поток вывода и сохранять их в свойстве **Накопленный стандартный вывод**.

Значение

«true»	Накопление стандартного вывода включено .
«false»	Накопление стандартного вывода отключено (по умолчанию).

2.9.3.14. Накопленный стандартный вывод

Доступно только для чтения в режиме рантайма.

```
String AccumulatedStdout
```

Предоставляет доступ к накопленному стандартному выводу «stdout» после выполнения процесса, запущенного с использованием компонента **Окружение: процесс**.

Если свойство **Накапливать стандартный вывод** установлено в значение «true», процесс накапливает стандартный вывод, а затем сохраняет его в данном свойстве.



ПРИМЕР

Вывести в текстовое поле накопленный стандартный вывод после завершения процесса.

```
Text_1.Text = Process_1.AccumulatedStdout;
```

2.9.3.15. Накапливать стандартный вывод ошибок

```
bool AccumulateStderr
```

Указывает, следует ли накапливать стандартный вывод ошибок «stderr» при выполнении процесса, запущенного с использованием компонента **Окружение: процесс**.

Когда данное свойство установлено в значение «true», компонент **Окружение: процесс** будет накапливать данные, выводимые в стандартный поток вывода ошибок и сохранять их в свойстве **Накопленный стандартный вывод ошибок**.

Значение

«true»	Накопление стандартного вывода ошибок включено .
«false»	Накопление стандартного вывода ошибок отключено (по умолчанию).

2.9.3.16. Накопленный стандартный вывод ошибок

Доступно только для чтения в режиме рантайма.

```
String AccumulatedStderr
```

Предоставляет доступ к накопленному стандартному выводу ошибок «stderr» после выполнения процесса, запущенного с использованием компонента **Окружение: процесс**.

Если свойство **Накапливать стандартный вывод ошибок** было установлено в «true» процесс накапливает стандартный вывод ошибок, а затем сохраняет его в данном свойстве.



ПРИМЕР

Вывести в текстовое поле накопленный стандартный вывод ошибок после завершения процесса.

```
Text_1.Text = Process_1.AccumulatedStderr;
```

2.10. Окружение: этот процесс

2.10.1. События

2.10.1.1. OnRunError

Ошибка запуска приложения.

Параметры

error	string	Описание ошибки, которая привела к неудачному запуску приложения.
-------	--------	-------------------------------------------------------------------

2.10.2. Функции

2.10.2.1. Run

Метод устарел. Для запуска процесса используйте компонент **Окружение : процесс**.

2.10.3. Свойства

2.10.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.10.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

2.10.3.3. Переменные среды процесса

Ссылка на компонент **Окружение**: **переменные среды процесса**.

Указание ссылки дает возможность использовать настройки переменных окружения, определенных в компоненте **Окружение**: **переменные среды процесса** при запуске процесса с помощью компонента **Окружение**: **процесс**. Если ссылка на **Окружение**: **переменные среды процесса** указана, но переменные окружения не настроены, то будут использоваться значения переменных окружения по умолчанию из операционной системы.

Примеры



ПРИМЕР

Записать в поле ввода значение переменной среды «TEMP»

```
TextEdit.Text = ThisProcess_1.Environment.Get("TEMP");
```

2.10.3.4. Рабочий каталог

```
string WorkingDirectory
```

Позволяет узнать рабочий каталог, в котором выполняется процесс SePlatform.HMI. Рабочий каталог представляет собой директорию, которая используется процессом SePlatform.HMI при запуске. Это означает, что все относительные пути к файлам и папкам, используемые процессом в ходе выполнения, будут относиться к этому рабочему каталогу.

Примеры



ПРИМЕР

Вывести текущий рабочий каталог процесса SePlatform.HMI в текстовое поле.

```
Text_1.Text = ThisProcess_1.WorkingDirectory;
```

2.10.3.5. ID

Доступно только для чтения в режиме рантайма.

```
UInt4 ID
```

ID процесса SePlatform.HMI.

Примеры



ПРИМЕР

Вывести ID процесса SePlatform.HMI в поле ввода.

```
TextEdit.Text = String.ToString(ThisProcess_1.ID);
```

2.10.3.6. Путь к исполняемому файлу процесса

Доступно только для чтения в режиме рантайма.

```
string ExecutablePath
```

Предоставляет информацию о местоположении исполняемого файла процесса SePlatform.HMI. Свойство может быть использовано для различных целей, таких как: проверка версии исполняемого файла, создание символьных ссылок на файл или выполнение дополнительных действий, зависящих от пути к исполняемому файлу.

Примеры



ПРИМЕР

Вывести в текстовое поле путь к исполняемому файлу процесса SePlatform.HMI

```
Text_2.Text = ThisProcess_1.ExecutablePath;
```

2.11. Окружение: переменные среды процесса

2.11.1. Свойства

2.11.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.11.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиначный объект
>«1»	массив соответствующей размерности

2.11.2. Функции

2.11.2.1. Get

Позволяет получить значение переменной среды по ее имени.

Синтаксис

```
string Get(string name);
```

Параметры

name	string	Имя переменной среды, значение которой необходимо получить.
------	--------	-------------------------------------------------------------

Возвращаемое значение

Возвращает значение переменной среды с указанным именем. Если переменная с таким именем не найдена, функция вернет пустую строку.

Примеры



ПРИМЕР

Вывести в текстовое поле значение переменной «TEMP»

```
Text_1.Text = ProcessEnvironment_1.Get("TEMP");
```

2.11.2.2. Set

Позволяет задать значение переменной среды по её имени. С помощью этой функции вы можете устанавливать значения как системных переменных среды, так и создавать или изменять пользовательские переменные, которые будут использоваться при запуске процессов с помощью компонента **Окружение : процесс**.


Синтаксис

```
void Set(string name, string value);
```

Параметры

name	string	Имя переменной среды, для которой нужно задать значение.
value	string	Значение, которое будет установлено для указанной переменной среды.

Примеры

 **ПРИМЕР**
Задать значение «Hello,World!» для пользовательской переменной «CUSTOM_VAR» .

```
ProcessEnvironment_1.Get("CUSTOM_VAR" , "Hello,World!");
```

2.11.2.3. Append

Используется для добавления значения в конец переменной среды-списка.

Если переменная среды уже содержит значения, то новое значение будет добавлено после последнего существующего значения с разделителем «;».

Если переменная среды пуста, то новое значение будет установлено как первое значение переменной.


Синтаксис

```
void Append(string name, string value);
```

Параметры

name	string	Имя переменной среды, к которой нужно добавить значение в конец переменной среды.
value	string	Значение, которое нужно добавить в конец переменной среды

Примеры

 **ПРИМЕР**
Для переменной среды «PATH» со значением C:\Program Files\SomeApp добавить в конец списка C:\NewFolder.

```
ProcessEnvironment_1.Append("PATH", "C:\\NewFolder");  
// Результат выполнения: переменная "PATH" станет "C:\Program  
Files\SomeApp;C:\NewFolder"
```

Если у нас нет переменной «PATH», и мы вызываем функцию Append("PATH", "C:\\NewFolder"), то после выполнения функции значение переменной «PATH» будет установлено как C:\NewFolder.

2.11.2.4. Prepend

Используется для добавления значения в начало переменной среды-списка.

Если переменная среды уже содержит значения, то новое значение будет добавлено перед существующим значением с разделителем «;».

Если переменная среды пуста, то новое значение будет установлено как первое значение переменной.

Синтаксис

```
void Prepend(string name, string value);
```

Параметры

name	string	Имя переменной среды, к которой нужно добавить значение в начало переменной среды.
value	string	Значение, которое нужно добавить в начало переменной среды.

Примеры



ПРИМЕР

Для переменной среды «PATH» со значением C:\Program Files\Java\jdk1.8.0\bin;C:\Python27;C:\Windows\System32 добавить в начало списка C:\MyApp.

```
ProcessEnvironment_1.Prepend("PATH", "C:\\MyApp");  
// Результат выполнения: переменная "PATH" станет "C:\\MyApp;C:\\Program  
Files\\Java\\jdk1.8.0\\bin;C:\\Python27;C:\\Windows\\System32"
```

Если у нас нет переменной «PATH», и мы вызываем функцию Prepend("PATH", "C:\\MyApp"), то после выполнения функции значение переменной «PATH» будет установлено как C:\MyApp.

2.11.2.5. Remove

Используется для удаления указанной переменной среды.

Синтаксис

```
void Remove(string name);
```

Параметры

name	string	Имя переменной среды, которую нужно удалить.
------	--------	----------------------------------------------

Примеры



ПРИМЕР

Удалить переменную среды «MY_VARIABLE».

```
ProcessEnvironment_1.Remove("MY_VARIABLE");
```

2.11.2.6. Clear

Используется для удаления всех переменных среды, которые были установлены в компоненте **Окружение: переменные среды процесса**.

Синтаксис

```
void Clear();
```

Примеры



ПРИМЕР

Удалить все переменные среды, которые были установлены в компоненте **Окружение: переменные среды процесса**.

```
ProcessEnvironment_1.Clear();
```

2.11.2.7. Reset

Используется для сброса компонента **Окружение: переменные среды процесса** к его первоначальному состоянию, удаляя все установленные переменные среды и возвращая компонент в исходное состояние.

Синтаксис

```
void Reset();
```

Примеры



ПРИМЕР

Сбросить компонент **Окружение: переменные среды процесса** к его первоначальному состоянию.

```
ProcessEnvironment_1.Reset();
```

2.12. Оборудование: клавиатура

Компонент позволяет получить текущую информацию о раскладке клавиатуры, состояниях клавиш и обрабатывать события смены раскладки.

2.12.1. Свойства

2.12.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.12.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

2.12.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

2.12.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

2.12.1.5. Язык ввода

string Locale

Текущая раскладка клавиатуры.

Примеры



ПРИМЕР

Вывести информацию о текущей раскладке клавиатуры в поле ввода.

```
TextEdit.Text = Keyboard.Locale;
```

2.12.2. Функции

2.12.2.1. LocalizedLanguageName

Получает имя языка ввода.

Примеры



ПРИМЕР

Вывести имя языка текущей раскладки клавиатуры в поле ввода.

```
TextEdit.Text = Keyboard.LocalizedLanguageName("en-US");
```

2.12.2.2. KeyState

Отслеживает текущее состояние клавиши клавиатуры.

Параметры

code	код клавиши
------	-------------

Возвращаемое значение

3	нажата и включена
2	отпущена и включена
1	нажата и выключена
0	отпущена и выключена

Коды клавиш

Цифры

Клавиша	Код	Клавиша	Код
0	0	5	5
1	1	6	6
2	2	7	7
3	3	8	8
4	4	9	9

Буквы

Клавиша	Код	Клавиша	Код	Клавиша	Код	Клавиша	Код	Клавиша	Код
A	10	G	16	M	22	S	28	Y	34
B	11	H	17	N	23	T	29	Z	35
C	12	I	18	O	24	U	30		
D	13	J	19	P	25	V	31		
E	14	K	20	Q	26	W	32		
F	15	L	21	R	27	X	33		

Символы

Клавиша	Код	Клавиша	Код
;	42	\	45
~`	36]	41
-_	37	'"	39
.>	44	,<	43
/	46	+	38
[40		

Модификаторы

Клавиша	Код
Shift	48
Ctrl	47
Alt	49
Mod	50
Caps Lock	78

Функциональные клавиши

Клавиша	Код	Клавиша	Код	Клавиша	Код	Клавиша	Код
F1	51	F7	57	F13	63	F19	69
F2	52	F8	58	F14	64	F20	70
F3	53	F9	59	F15	65	F21	71
F4	54	F10	60	F16	66	F22	72
F5	55	F11	61	F17	67	F23	73
F6	56	F12	62	F18	68	F24	74

Цифровая клавиатура

Клавиша	Код	Клавиша	Код
NumPad 0	86	NumPad 8	94
NumPad 1	87	NumPad 9	95
NumPad 2	88	NumPad *	97
NumPad 3	89	NumPad +	99
NumPad 4	90	NumPad -	98
NumPad 5	91	NumPad .	100
NumPad 6	92	NumPad /	96
NumPad 7	93		

Управляющие клавиши

Клавиша	Код	Клавиша	Код
Esc	75	Num Lock	79
Tab	85	Del	109
Pause	80	Scroll Lock	77
Backspace	83	Win	50
Enter	82	Menu	81
PrtSc	76	Space	84
Ins	110		

Клавиши навигации

Клавиша	Код	Клавиша	Код
PageUp	103	←	108
PageDown	104	↑	105
End	102	→	106
Home	101	↓	107

2.13. Оборудование: монитор

Компонент позволяет взаимодействовать с монитором или группой мониторов многомониторной системы отображения.

2.13.1. Свойства

2.13.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

2.13.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиначный объект
>«1»	массив соответствующей размерности

2.13.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

2.13.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

2.13.2. Функции

2.13.2.1. Count

Отображает количество мониторов на рабочем месте.

Примеры

**ПРИМЕР**

Отобразить в поле ввода количество мониторов на рабочем месте.

```
TextEdit.Text = MonitorList.Count();
```

2.13.2.2. GetResolutionX-GetResolutionY

Вычисляют разрешение монитора с номером Number по горизонтали и вертикали в пикселях. Входной параметр - номер монитора (Number). Нумерация мониторов начинается с 0 (0 - это крайний левый верхний монитор) и не зависит от нумерации мониторов в операционной системе. Направление нумерации происходит «сверху-вниз» и «слева-направо».

На рисунке ниже показаны системные номера мониторов. В SePlatform.HMI данным мониторам будут соответствовать следующие номера:

- Монитор 2: Number = 0;
- Монитор 1: Number = 1;
- Монитор 3: Number = 2.



Примеры



ПРИМЕР

Вывести в поля ввода разрешение монитора под номером 2.

```
TextEdit1.Text = MonitorList.GetResolutionX(0);  
TextEdit2.Text = MonitorList.GetResolutionY(0);
```

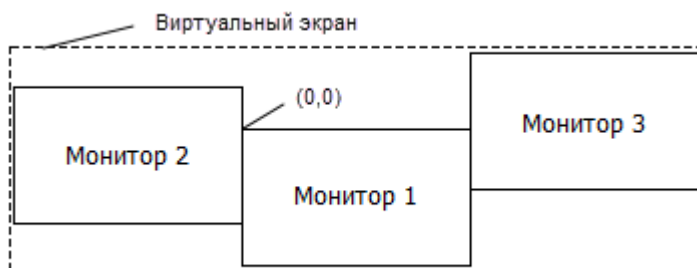
2.13.2.3. GetX-GetY

Вычисляют абсолютную координату монитора с номером Number относительно многомониторной системы отображения. Входной параметр - номер монитора (Number). Нумерация мониторов начинается с 0 (0 - это крайний левый верхний монитор) и не зависит от нумерации мониторов в операционной системе. Направление нумерации происходит «сверху-вниз» и «слева-направо».

Первый монитор в операционной системе содержит начало координат (0,0).

На рисунке ниже показаны системные номера мониторов. В SePlatform.HMI данным мониторам будут соответствовать следующие номера:

- Монитор 2: Number = 0;
- Монитор 1: Number = 1;
- Монитор 3: Number = 2.



Примеры



ПРИМЕР

Вывести координаты монитора 2 относительно многомониторной системы отображения.

```
TextEdit1.Text = MonitorList.GetX(0);  
TextEdit2.Text = MonitorList.GetY(0);
```

3. Компоненты для предоставления сервисных функций

3.1. Узел статистики ApService

3.1.1. События

3.1.1.1. NodeStateChanged

Обновление данных об узле, на который оформлена подписка.

Параметры

path	string	Полный путь узла статистики, по которому произошло событие. Значение «Path» должно передаваться в экранированном виде.
value	variant	Актуальное значение узла статистики.
cdt	uint1	Тип актуального значения узла статистики. <ul style="list-style-type: none"> ➤ «0» - empty, ➤ «1» - bool, ➤ «2» - int1, ➤ «3» - uint1, ➤ «4» - int2, ➤ «5» - uint2, ➤ «6 »- int4, ➤ «7» - uint4, ➤ «8» - int8, ➤ «9» - uint8, ➤ «10» - float, ➤ «11» - double, ➤ «12 »- timestamp (uint8), ➤ «13 »- UUID (16 байт), ➤ «14» - string
time	uint8	Метка времени последней актуализации состояния узла.
nodeType	uint1	Актуальный тип узла статистики. <ul style="list-style-type: none"> ➤ «0» - неизвестный тип, ➤ «1» - папка, ➤ «2» - секция, ➤ «3» - значение.

nodeStatus	uint1	<p>Актуальный статус узла статистики.</p> <ul style="list-style-type: none"> ➤ «0» - информация о существовании узла по заданному пути еще не была получена, ➤ «1» - узел существует на источнике, ➤ «2» - узел не существует на источнике.
------------	-------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Примеры



ПРИМЕР

Используя параметры «value» и «cdt» события **NodeStateChanged** вывести в текстовое поле актуальное значение и тип узла статистики.

```
Text_1.Text = VariantToString(value, cdt);
```

3.1.1.2. PollingFailed

Уведомление об ошибке при попытке совершения запроса для актуализации состояния узла, на который оформлена подписка.

Параметры

path	string	Путь узла статистики, по которому произошло событие.
errorDescription	string	Текстовое описание ошибки.

3.1.2. Свойства

3.1.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

3.1.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

3.1.2.3. Источник статистики ApService

Source

Источник статистики, поставляющий статистические данные. Указывается в виде ссылки на объект типа **Источник статистики ApService**, существующий в проекте.

3.1.2.4. Активность подписки

bool Active

Управляет активностью подписки на изменения отслеживаемого узла статистики.

Значение

«true»	Компонент будет инициировать запросы к Источнику данных статистики ApService для получения обновлений.
«false»	Компонент не будет инициировать запросы к Источнику данных статистики ApService .

3.1.2.5. Период опроса

UInt4 PollingPeriod

Устанавливает временной интервал между опросами значений отслеживаемого узла статистики по подписке.

Примеры



ПРИМЕР

Задать период опроса отслеживаемого узла статистики через каждые 5 секунд.

```
ApStatItem_1.PollingPeriod = 5000;
```

3.1.2.6. Путь

string Path

Полное имя узла статистики, которое используется для оформления подписки на изменения значений этого узла.

**ПРИМЕЧАНИЕ**

Значение свойства **Путь** должно быть указано в экранированной форме: все символы, которые совпадают с символами разделителей уровней иерархии дерева узлов статистики, но по смыслу не являются таковыми, должны быть экранированы.

Примеры

**ПРИМЕР**

Укажите путь "папка1.общие параметры.соединение с сеплатформ.сервер" в экранированной форме для оформления подписки:

```
ApStatItem_1.Path("папка1.общие параметры.соединение с сеплатформ\\.сервер");
```

Два обратных следа «\» перед точкой «.» используются для того, чтобы указать, что точка «.» является не разделителем уровней в иерархии узлов, а просто частью имени узла.

3.2. Json-браузер источника статистики ApService

3.2.1. События

3.2.1.1. GetChildrenComplete

Уведомление об успешном завершении браузинга узла статистики.

Параметры

path	string	Путь узла статистики, браузеринг которого был выполнен.
requestId	uint4	Идентификатор запроса.
result	string	JSON-строка с данными об узлах статистики, по которым был осуществлен запрос, согласно заданным свойствам компонента Json-браузер источника статистики ApService .

3.2.1.2. GetChildrenFailed

Уведомление об ошибке браузинга узла статистики.

Параметры

path	string	Путь узла статистики, браузеринг которого был запрошен.
------	--------	---------------------------------------------------------

requestId	uint4	Идентификатор запроса.
errorDescription	string	Текстовое описание ошибки браузинга узла статистики.

3.2.1.3. ResolvePathComplete

Уведомление об успешном завершении разбора пути узла статистики.

Параметры

path	string	Путь узла статистики, разбор которого был выполнен.
requestId	uint4	Идентификатор запроса.
errorDescription	string	JSON-строка с информацией о положении узла в иерархии дерева статистики.

3.2.1.4. ResolvePathFailed

Уведомление об ошибке разбора пути узла статистики.

Параметры

path	string	Путь узла статистики, разбор которого был запрошен.
requestId	uint4	Идентификатор запроса.
errorDescription	string	Текстовое описание ошибки разбора пути узла статистики.

3.2.2. Функции

3.2.2.1. GetChildren

Метод предназначен для выполнения браузинга узлов статистики и формирования среза дерева в формате JSON на основе заданных свойств у компонента **Json-браузер источника статистики ApiService**.

Результаты браузинга следует получать через события **GetChildrenComplete** и **GetChildrenFailed**. JSON-строка, полученная после успешного браузинга, будет содержать данные об узлах дерева статистики в соответствии с заданными свойствами у компонента **Json-браузер источника статистики ApiService**.

Синтаксис

```
uint4 GetChildren(string path)
```

Параметры

path	string	Путь к узлу статистики, для которого требуется выполнить браузеринг. Путь должен быть представлен в экранированном виде.
------	--------	--------------------------------------------------------------------------------------------------------------------------

Возвращаемое значение

requestId	uint4	Целочисленный идентификатор запроса.
-----------	-------	--------------------------------------

Примеры



ПРИМЕР

Выполнить браузеринг узла дерева статистики, начиная с пути папка1.общие параметры.сервер. Полученный JSON вывести в текстовое поле.

```
ApStatJsonBrowser_1.GetChildren("папка1.общие параметры.сервер");
```

В обработке события **GetChildrenComplete** у компонента **Json-браузер источника статистики ApiService** пропишите следующий код:

```
Text_1.Text = result;
```

3.2.2.2. ResolvePath

Метод предназначен для разбора пути узла статистики с формированием json-строки, содержащей информацию о положении узла в иерархии дерева статистики.

Результаты разбора следует ожидать через события **ResolvePathComplete** и **ResolvePathFailed**. JSON-строка, полученная после успешного разбора, будет содержать информацию об иерархии узлов в виде массива объектов, представляющих узлы и их характеристики.

Синтаксис

```
uint4 ResolvePath(string path)
```

Параметры

path	string	Путь к узлу статистики, для которого требуется выполнить разбор. Путь должен быть представлен в экранированном виде.
------	--------	----------------------------------------------------------------------------------------------------------------------

Возвращаемое значение

requestId	uint4	Целочисленный идентификатор запроса.
-----------	-------	--------------------------------------

Примеры



ПРИМЕР

Выполнить разбор пути узла статистики Модули.TCP Server Module.Общие параметры.Идентификатор модуля. Результат вывести в текстовое поле.

```
ResolvePath("Модули.TCP Server Module.Общие параметры.Идентификатор модуля");
```

В обработке события **ResolvePathComplete** пропишите следующий код:

```
ResolveResult.Text = result;
```

3.2.3. Свойства

3.2.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

3.2.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

3.2.3.3. Источник статистики ApService

Source

Источник статистики, поставляющий статистические данные. Указывается в виде ссылки на объект типа **Источник статистики ApService**, существующий в проекте.

3.2.3.4. Включение в результат брауза узлов-папок

```
bool IncludeFolders
```

Включает в результат брауза узлов статистики узлы типа "папка".

Значение

«true»	Узлы-папки будут включены в результаты брауза.
«false»	Узлы-папки не будут включены в результаты брауза.

3.2.3.5. Включение в результат брауза узлов-секций

```
bool IncludeSections
```

Включает в результат брауза узлов статистики узлы типа "секции".

Значение

«true»	Узлы-секции будут включены в результаты брауза.
«false»	Узлы-секции не будут включены в результаты брауза.

3.2.3.6. Включение в результат брауза узлов-значений

```
bool IncludeItems
```

Включает в результат брауза узлов статистики узлы типа "параметры".

Значение

«true»	Узлы-параметры будут включены в результаты брауза.
«false»	Узлы-параметры не будут включены в результаты брауза.

3.2.3.7. Сохранять в результат брауза тип значения узла

```
bool IncludeNodeValueType
```

Включает в результат брауза узлов статистики информацию о типе значений узлов.

Значение

«true»	Тип значения узла будет включен в результат брауза
«false»	Тип значения узла не будет включен в результат брауза

3.2.3.8. Сохранять в результат брауза значения узлов

```
bool IncludeNodeValue
```

Включает в результат брауза узлов статистики информацию о значении узлов.

Значение

«true»	Значение узла будет включено в результат брауза.
«false»	Значение узла не будет включено в результат брауза.

3.2.3.9. Рекурсивный брауз

```
bool Recursive
```

Определяет, будет ли компонент **Json-браузер источника статистики ApiService** получать данные об узлах дерева статистики на всех уровнях иерархии дерева статистики или только для первого уровня.

Значение

«true»	Json-браузер источника статистики ApiService будет работать в рекурсивном режиме. В этом режиме браузер будет получать данные об узлах на всех уровнях иерархии дерева статистики, включая все подуровни.
«false»	Json-браузер источника статистики ApiService будет работать в инкрементальном режиме. В этом режиме браузер будет получать данные только для первого уровня узлов дерева статистики, не включая данные из более глубоких уровней.

3.2.3.10. Форматирование json-строки результата брауза в читабельном виде

```
bool PrettyFormat
```

Определяет режим отображения результатов браузера в виде JSON-строки.

Значение

«true»	Отображение результатов в читабельном виде с отступами и переносами строк.
«false»	Компактное отображение результатов, без лишних пробелов и переносов строк.

3.3. Rmap-браузер источника статистики ApService

3.3.1. Функции

3.3.1.1. GetChildren

При вызове этой функции компонент **Rmap-браузер источника статистики ApService** выполняет браузинг узлов дерева статистики по указанному пути.

Пустая строка в качестве пути означает начало браузинга с корня дерева статистики.



ПРИМЕЧАНИЕ

Переданный путь должен быть в экранированном виде и соответствовать структуре дерева статистики. Все символы, которые совпадают с символами разделителей уровней иерархии дерева узлов статистики, но по смыслу не являются таковыми, должны быть экранированы.

Синтаксис

```
void GetChildren(string path)
```

Параметры

path	string	Путь к узлу статистики для осуществления брауза.
------	--------	--------------------------------------------------

Примеры



ПРИМЕР

Выполнить браузинг узлов статистики, начиная с корня дерева статистики.

```
ApStatRmapBrowser_1.GetChildren(" ");
```

**ПРИМЕР**

Выполнить браузеринг узла статистики, начиная с пути папка1.общие параметры.соединение с сеплатформ.сервер. Путь передать в экранированной форме.

```
ApStatRmapBrowser_1.GetChildren("папка1.общие параметры.соединение с
сеплатформ\\.сервер");
```

Два обратных следа «\\» перед точкой «.» используются для того, чтобы указать, что точка «.» является не разделителем уровней в иерархии узлов, а просто частью имени узла.

3.3.1.2. GetChildrenWithRequestId

При вызове этой функции компонент **Rmap-браузер источника статистики ApService** создает запрос к источнику данных статистики для получения узлов дерева статистики по указанному пути.

Результаты запроса (дочерние узлы) автоматически прикрепляются к узлу с идентификатором «**requestId**» в графическом компоненте **Дерево** (или другой структуре данных), который был создан заранее с использованием метода **AddItem()**.

Синтаксис

```
void GetChildrenWithRequestId(string path, string requestId)
```

Параметры

path	string	Путь к узлу статистики, для которого будет выполнен браузер.
requestId	string	Идентификатор запроса, который соответствует родительскому узлу, к которому должен быть прикреплен результат браузера.

Примеры

**ПРИМЕР**

Создать корневой элемент «**Модули**» в компоненте **Дерево** и загрузить дочерние элементы для узла «**Модули**».

```
x: string = Tree_1.AddItem("", "Модули");
```

Вызываем метод **GetChildrenWithRequestId()** для загрузки дочерних элементов для узла «**Модули**» с использованием идентификатора «**x**»:

```
ApStatRmapBrowser_1.GetChildrenWithRequestId("Модули", x);
```

3.3.2. Свойства

3.3.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

3.3.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

3.3.2.3. Источник статистики ApService

Source

Источник статистики, поставляющий статистические данные. Указывается в виде ссылки на объект типа **Источник статистики ApService**, существующий в проекте.

3.3.2.4. Брауз узлов-папок

bool IncludeFolders

Включает в результат брауза узлов статистики узлы типа "папка".

Значение

«true»	Узлы-папки будут включены в результаты брауза.
«false»	Узлы-папки не будут включены в результаты брауза.

3.3.2.5. Брауз узлов-секций

bool IncludeSections

Включает в результат брауза узлов статистики узлы типа "секции".

Значение

«true»	Узлы-секции будут включены в результаты брауза.
«false»	Узлы-секции не будут включены в результаты брауза.

3.3.2.6. Брауз узлов-значений

```
bool IncludeItems
```

Включает в результат брауза узлов статистики узлы типа "параметры".

Значение

«true»	Узлы-параметры будут включены в результаты брауза.
«false»	Узлы-параметры не будут включены в результаты брауза.

3.3.2.7. Рекурсивный брауз

```
bool Recursive
```

Определяет, будет ли компонент **Rmap-браузер источника статистики ApService** получать данные об узлах дерева статистики на всех уровнях иерархии дерева статистики или только для первого уровня.

Значение

«true»	Rmap-браузер источника статистики ApService будет работать в рекурсивном режиме. В этом режиме браузер будет получать данные об узлах на всех уровнях иерархии дерева статистики, включая все подуровни.
«false»	Rmap-браузер источника статистики ApService будет работать в инкрементальном режиме. В этом режиме браузер будет получать данные только для первого уровня узлов дерева статистики, не включая данные из более глубоких уровней.

3.4. Источник данных статистики ApService

3.4.1. События

3.4.1.1. OnConnectionStateChanged

Изменение состояния подключения к источнику данных статистики.

Параметры

connectionState	uint1	Состояние подключения к источнику данных статистики. <ul style="list-style-type: none">➤ «0» – нет подключения;➤ «1» – выполняется подключение;➤ «2» – подключено;➤ «3» – выполняется отключение.
errorState	uint1	Состояние ошибки подключения к источнику данных статистики. <ul style="list-style-type: none">➤ «0» – нет ошибки;➤ «1» – таймаут подключения к серверу статистики (не используется);➤ «2» – ошибка авторизации на сервере статистики;➤ «3» – ошибка подключения к серверу статистики;➤ «4» – ошибка обработки документа среза статистики;➤ «5» – документ среза статистики имеет некорректный формат (схему);➤ «6» – формат документа, указанный в свойстве Формат среза дерева статистики, не поддерживается.

Примеры

 **ПРИМЕР**
Вывести в текстовое поле состояние соединения с источником данных статистики.

```
Text_1.Text = String.toString(connectionState);
```

3.4.2. Функции

3.4.2.1. EscapePathPart

Используется для экранирования символов, совпадающих с символом разделения уровней иерархии узлов (например, точки) в сокращенных именах узлов статистики. Это необходимо для обеспечения корректной передачи полных путей к узлам статистики, особенно в случае, когда сокращенные имена содержат символы, которые могут быть неправильно интерпретированы в составе полного имени.

Синтаксис

```
string EscapePathPart(string pathPart)
```

Параметры

pathPart	string	Часть полного пути узла, соответствующая одному уровню иерархии дерева статистики, которую необходимо экранировать.
----------	--------	---------------------------------------------------------------------------------------------------------------------

Возвращаемое значение

Возвращает экранированную версию сокращенного имени узла, в которой символы разделения уровней заменены на безопасные символы.

Примеры



ПРИМЕР

Вывести в текстовое поле экранированную и исходную версию сокращенного имени узла.

```
x: var = "папка1.общие параметры.соединение с сеплатформ.сервер"; // исходный путь
y: var = ApStatSource_1.EscapePathPart("папка1") + "." +
  ApStatSource_1.EscapePathPart("общие параметры") + "." +
  ApStatSource_1.EscapePathPart("соединение с сеплатформ.сервер"); // экранированный
путь

Text_1.Text = "Исходный путь: " + x; // Вывод исходного пути в текстовое поле
Text_2.Text = "Экранированный путь: " + y; // Вывод экранированного пути в текстовое
поле
```

Результат выполнения:

```
Исходный путь: папка1.общие параметры.соединение с сеплатформ.сервер
Экранированный путь: папка1.общие параметры.соединение с сеплатформ\сервер
```

3.4.2.2. GetConnectionStateAsString

Позволяет получить строковое представление состояния подключения к источнику данных статистики на основе параметра `connectionState`, передаваемого из события `OnConnectionStateChanged`.

Синтаксис

```
string GetConnectionStateAsString(uint1 connectionState)
```

Параметры

connectionState	uint1	Идентификатор состояния подключения к источнику данных статистики.
-----------------	-------	--------------------------------------------------------------------

Возвращаемое значение

Возвращает строку, представляющую текущее состояние подключения к источнику данных статистики.

Примеры



ПРИМЕР

Используя параметр `connectionState` события **OnConnectionStateChanged** вывести в текстовое поле текущее состояние подключения к источнику данных статистики.

```
ApServiceSourceConnectionState.Text = ApServiceStatsSource.GetConnectionStateAsString(connectionState);
```

3.4.2.3. GetErrorStateAsString

Позволяет получить строковое представление состояния ошибки подключения к источнику данных статистики на основе параметра `errorState`, передаваемого из события **OnConnectionStateChanged**.

Синтаксис

```
string GetErrorStateAsString(uint1 errorState)
```

Параметры

errorState	uint1	Идентификатор состояния ошибки подключения к источнику данных статистики.
------------	-------	---------------------------------------------------------------------------

Возвращаемое значение

Возвращает строку, представляющую текущее состояние ошибки подключения к источнику данных статистики.

Примеры



ПРИМЕР

Используя параметр `errorState` события `OnConnectionStateChanged` вывести в текстовое поле текущее состояние ошибки подключения к источнику данных статистики.

```
ApServiceSourceConnectionState.Text = ApServiceStatsSource.GetErrorStateAsString(errorState);
```

3.4.3. Свойства

3.4.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

3.4.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

3.4.3.3. Источник учетных данных

Credentials

Используется при подключении к серверу, который защищен паролем. Указывается в виде ссылки на объект типа **Учетные данные AP**, существующий в проекте.

3.4.3.4. Имя клиента

string ClientDisplayName

Позволяет установить отображаемое на сервере имя текущего клиента при подключении. Это имя передается на сервер и может быть видимо для других клиентов, чтобы идентифицировать текущего подключившегося клиента.

Примеры



ПРИМЕР

Вывести в текстовое поле имя текущего клиента, отображаемое при подключении.

```
LineEdit_1.Text = ApiServiceStatsSource.ClientDisplayName;
```

3.4.3.5. Режим работы источника

UInt1 Mode

Используется для выбора способа, которым компонент **Источник данных статистики ApiService** будет получать статистические данные.

Значение

0	В этом режиме компонент Источник данных статистики ApiService будет получать статистические данные напрямую с сервера. Используйте этот режим при стабильном соединении с сервером для получения данных в реальном времени.
1	В этом режиме предполагается использование среза дерева статистики. Чтобы начать работу в этом режиме, необходимо заполнить свойство Срез дерева статистики , содержащее данные в формате JSON или XML, представляющие срез дерева статистики. Используйте этот режим, если у вас нет доступа к серверу, но есть сохраненные данные о статистике.

3.4.3.6. Таймаут операций получения данных

uint4 Timeout

Позволяет установить время ожидания ответа от сервера. Этот параметр определяет, сколько времени компонент **Источник данных статистики ApiService** будет ждать ответа от сервера перед тем, как считать операцию получения данных неудачной.

3.4.3.7. Хост

string Location

Позволяет указать сетевой адрес (хост) сервера, с которым будет устанавливаться соединение.

3.4.3.8. Порт

```
uint2 Port
```

Позволяет задать порт сервера, через который будет осуществляться соединение.

По умолчанию для подключения к серверам используются следующие порты:

- SePlatform.Data Server - 4572;
- SePlatform.AccessPoint - 4976;
- SePlatform.Historian - 3388;
- SePlatform.Imitator - 4983;
- SePlatform.License Server - 15151.

3.4.3.9. Срез дерева статистики

```
string SnapShot
```

Свойство хранит срез дерева статистики в формате JSON или XML. Срез представляет собой массив JSON-объектов или структуру XML. Каждый элемент массива JSON или XML представляет узел статистики, содержащий информацию о его имени, типе, значении и дочерних узлах.

Для использования источника статистики в режиме среза, создайте необходимую структуру в формате XML или JSON и вставьте её содержимое в данное свойство. Способы создания среза описаны в разделе Как создать срез дерева статистики.

- Пример среза в формате XML

```
<StatisticFolder Name="File">
  <Folders>
    <StatisticFolder Name="Приложение">
      <Attributes>
        <StatisticAttribute Name="Имя экземпляра приложения Value="SePlatform.License
Server.Agent" />
        <StatisticAttribute Name="Версия" Value="1.11.1.1" />
        <StatisticAttribute Name="Версия старта" Value="21.09.2023 06:12:34" />
      </Attributes>
    </Folders>
    <StatisticFolder Name="Клиенты">
      <Folders>
        <StatisticFolder Name="Клиент 2">
          <Attributes>
            <StatisticAttribute Name="Адрес клиента" Value="127.0.0.1:49675" />
            <StatisticAttribute Name="Время подключения " Value="21.09.2023 06:12:35" />
          </Attributes>
        </StatisticFolder>
        <StatisticAttribute Name="Идентификатор сессии" Value="1" />
        <StatisticAttribute Name="Количество операций" Value="379395" />
      </Attributes>
    </StatisticFolder>
  </Folders>
</StatisticFolder>
```

```
        </Folders>
    </StatisticFolder>
</Folders>
</StatisticFolder>
</Folders>
</StatisticFolder>
```

«**<StatisticFolder>**» - папка в структуре статистики.

«**<Folders>**» - список подпапок (позволяет организовать их вложенность).

«**<StatisticAttribute>**» - атрибут с указанным именем.

«**<Value>**» - значение атрибута.

➤ Пример среза в формате JSON

```
[
  {
    "Name": "Приложение",
    "Type": 1,
    "ValueType": 0,
    "Value": null,
    "Children": [
      {
        "Name": "Имя экземпляра приложения",
        "Type": 3,
        "ValueType": 14,
        "Value": "SePlatform.License Server.Agent",
        "Children": []
      },
      {
        "Name": "Версия",
        "Type": 3,
        "ValueType": 14,
        "Value": "1.11.1.1",
        "Children": []
      },
      {
        "Name": "Время старта",
        "Type": 3,
        "ValueType": 14,
        "Value": "21.09.2023 06:12:34",
        "Children": []
      },
      {
        "Name": "Клиенты",
        "Type": 1,
        "ValueType": 0,
        "Value": null,
        "Children": [
          {
            "Name": "Клиент 2",
            "Type": 1,
            "ValueType": 0,
            "Value": null,
            "Children": [
              {
                "Name": "Адрес клиента",
                "Type": 3,
                "ValueType": 14,
                "Value": "127.0.0.1:49675",
                "Children": []
              },
              {
                "Name": "Время подключения",
                "Type": 3,
```

```
        "ValueType": 14,
        "Value": "21.09.2023 06:12:35",
        "Children": []
      },
      {
        "Name": "Идентификатор сессии",
        "Type": 3,
        "ValueType": 14,
        "Value": "1",
        "Children": []
      },
      {
        "Name": "Количество операций",
        "Type": 3,
        "ValueType": 7,
        "Value": 380045,
        "Children": []
      }
    ]
  }
}
```

- «Name» - имя узла.
- «Type» - идентификатор типа узла.
- «ValueType» - числовой идентификатор типа значения.
- «Value» - значение данного элемента.
- «Children» - массив, содержащий дочерние элементы данного узла.

3.4.3.10. Формат среза дерева статистики

```
string SnapShotFormat
```

Свойство позволяет указать формат, в котором представлены данные о дереве статистики , переданные через свойство [Срез дерева статистики](#).

Значения

«json»	Срез дерева статистики предоставлен в формате JSON.
«xml»	Срез дерева статистики предоставлен в формате XML.

3.4.3.11. Active

bool Active

Включает/выключает активность компонента.

Значения

«true»	Компонент активен.
«false»	Компонент неактивен

4. Запрос SQL

Компонент **Запрос SQL** предназначен для выполнения SQL-запросов к базе данных ([PostgreSQL](#), [MySQL](#), [Microsoft SQL Server](#) и другие).

4.1. События

4.1.1. OnOperationFinished

Завершение выполнения запроса.

Параметры

operationType	uint1	Тип операции. <ul style="list-style-type: none">➤ «0» - Подготовка запроса;➤ «1» - Выполнение запроса.
operationResult	uint1	Результат выполнения операции. <ul style="list-style-type: none">➤ «0» - Операция успешно выполнена;➤ «1» - При выполнении операции произошла ошибка.
message	string	Текст сообщения по результатам выполнения операции.



ПРИМЕР

Используя параметры «operationResult» и «message» события **OnOperationFinished** вывести в текстовое поле результат выполнения SQL-запроса.

```
if (operationType == 1) {  
    if ( operationResult == 0 )  
        Text_1.Text = "Success:" + message;  
    else  
        Text_1.Text = "Error" + message;  
}
```

4.1.2. OnConnectionStateChanged

Изменение состояния подключения к базе данных.

Параметры

state	uint1	Состояние соединения. <ul style="list-style-type: none">➤ «0» - Соединение не установлено;➤ «1» - Соединение установлено.
-------	-------	--------------------------------------------------------------------------------------------------------------------------------------------------

stateAsString	string	Состояние соединения в строковом представлении.
---------------	--------	-------------------------------------------------

**ПРИМЕР**

Используя параметры «state» и «stateAsString» события **OnConnectionStateChanged** вывести в текстовое поле состояние соединения с базой данных. В случае отсутствия соединения вывести уведомление об ошибке.

```
if ( state == 0 )
    Text_1.Text = "ConnectError:" + stateAsString;
else
    Text_1.Text = "Connected!";
```

4.2. Функции

4.2.1. Connect

Устанавливает соединение с базой данных.

4.2.1.1. Синтаксис

```
void Connect();
```

4.2.2. Execute

Выполняет SQL-запрос, переданный через свойство **Текст запроса** ([стр. 206](#)).

4.2.2.1. Синтаксис

```
void Execute();
```

4.2.3. Prepare

Выполняет подготовку SQL-запроса, переданного через свойство **Текст запроса** ([стр. 207](#)). Подготовка включает в себя синтаксическую проверку SQL-запроса, построение плана выполнения запроса и выделение под него вычислительных ресурсов.

4.2.3.1. Синтаксис

```
void Prepare();
```

4.2.4. Disconnect

Разрывает соединение с базой данных. Используется для освобождения системных ресурсов, занятых подключением к базе данных, и предотвращения утечек памяти.

4.2.4.1. Синтаксис

```
void Disconnect();
```

4.3. Свойства

4.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

4.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	Одиночный объект.
>«1»	Массив соответствующей размерности.

4.3.3. Строка подключения

```
string connectionString
```

Строка, которая содержит информацию, необходимую ODBC драйверу для подключения к базе данных.

В примере ниже используется строка подключения, которая содержит информацию о драйвере «Driver» для PostgreSQL, адресе сервера «Server», порте «Port», базе данных «Database», а также учетные данные пользователя «Uid - имя пользователя, Pwd - пароль».

```
Driver={PostgreSQL};Server=localhost;Port=5432;Database=postgres;Uid=postgres;Pwd=password
```

Вместо использования целого ряда атрибутов, определяющих способ подключения драйвера к источнику данных, можно указать только имя системного источника данных (DSN):

```
DSN=PostgreSQL35W
```

**ПРИМЕЧАНИЕ**

Системные источники данных можно посмотреть в Панель управления → Администрирование → Администратор источника данных ODBC → Вкладка Системный DSN.

Примеры

**ПРИМЕР**

Настроить подключение к БД, если имя системного источника данных «PostgreSQL35W»

```
SqlQuery_1.ConnectionString = "DSN=PostgreSQL35W";
```

4.3.4. Текст запроса

```
string Text
```

Текст SQL-запроса.

Примеры

**ПРИМЕР**

Выполнить «SELECT»запрос для выборки данных по всем столбцам из таблицы «Numbers»:

```
Table_1.TableModel_1.SqlQuery_1.Text = "SELECT * FROM Numbers";
```

4.3.5. Состояние подключения к БД

```
uint1 ConnectionState
```

Текущий статус соединения с базой данных.

**ПРИМЕЧАНИЕ**

Доступно только для чтения в режиме рантайма.

Значение

«1»	Соединение установлено.
«0»	Соединение не установлено.

Примеры



ПРИМЕР

Подключиться к базе данных, если соединение с ней не установлено.

```
if ( Table_1.TableModel_1.SqlQuery_1.ConnectionState == 0 )  
    Table_1.TableModel_1.SqlQuery_1.Connect( );
```


5. Компоненты печати

5.1. Печать

5.1.1. Функции

5.1.1.1. Print()

Отправляет на печать установленную область, заданную в свойстве **Область печати**.

Активирует процесс печати немедленно, без предварительного предпросмотра или изменения настроек печати.

Синтаксис

```
void Print();
```

Примеры



ПРИМЕР

Отправить на печать выбранную область, указанную в свойстве **Область печати**

```
Print_1.Print();
```

5.1.1.2. PrintDialog()

Открывает диалоговое окно печати, предоставляя пользователю возможность настроить параметры печати перед отправкой на принтер.

Синтаксис

```
void PrintDialog()
```

Примеры



ПРИМЕР

Открыть диалоговое окно печати для настройки параметров печати

```
Print_1.PrintDialog();
```


5.1.1.3. PrintPreviewDialog()

Открывает диалоговое окно предпросмотра печати, позволяя пользователю просмотреть, как будет выглядеть печатаемая область перед её фактической печатью.

Синтаксис

```
void PrintPreviewDialog()
```

Примеры

 ПРИМЕР

Открыть диалога предпросмотра печати для настройки и предварительного просмотра

```
Print_1.PrintPreviewDialog();
```

5.1.2. Свойства

5.1.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

5.1.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

5.1.2.3. Принтер

```
Printer
```

Ссылка на объект типа **Принтер**, существующий в проекте.

5.1.2.4. Область печати

uint1 Area

Область графического интерфейса, которая будет отправлена на печать.

Значение

«Родительский элемент»	Печатается графический элемент, в котором находится экземпляр компонента Печать .
«Форма»	Печатается содержимое родительского контекста, в рамках которого размещён компонент Печать . Это может быть, например, форма во фрейме или форма в окне.
«Содержимое окна»	Печатается содержимое самого корневого по отношению к окну контекста, что обычно является формой окна.

5.1.2.5. Автоподбор ориентации

bool AutoPageOrientation

Автоподбор ориентации страницы при печати.

Значение

«true»	Автоподбор ориентации включен. Система автоматически выбирает ориентацию страницы на основе содержимого и размеров печатаемой области.
«false»	Автоподбор ориентации выключен. Ориентация страницы не будет изменяться автоматически, она останется такой, как была задана вручную.

5.2. Принтер

5.2.1. Свойства

5.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

5.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение


«1»	одиночный объект
>«1»	массив соответствующей размерности

5.2.1.3. Имя принтера

```
string Name
```

Имя принтера, который будет использоваться для печати. Пустая строка означает выбор системного принтера по умолчанию.

Примеры

 ПРИМЕР

Вывести в текстовое поле имя принтера, который выбран для печати

```
Text_1.Text = Printer_1.Name; // Отображает имя выбранного принтера
```

5.2.1.4. Цветовой режим

```
uint1 ColorMode
```

Режим печати в отношении цвета.

Значение

«0»	Черно-белая печать.
«1»	Цветная печать.

Примеры



ПРИМЕР

Настроить принтер на черно-белую печать

```
Printer_1.ColorMode = 0; // Настройка принтера на черно-белую печать
```

5.2.1.5. Количество копий

```
int4 CopyCount
```

Количество копий документа, которое будет напечатано.

5.2.1.6. Ориентация страницы

```
uint1 PageOrientation
```

Ориентация страницы для печати.

Значение

«0»	Книжная ориентация. Это вертикальная ориентация страницы, где высота страницы больше её ширины.
«1»	Альбомная ориентация. Это горизонтальная ориентация страницы, где ширина страницы больше её высоты.

5.2.1.7. Качество печати

```
int4 Resolution
```

Качество печати, выраженное через разрешение в DPI (точек на дюйм). Большее значение DPI обеспечивает более высокое качество печати, но может увеличить время печати и расход материалов.

Значение

«0»	Используется разрешение принтера по умолчанию. Принтер автоматически выбирает оптимальное разрешение в зависимости от своих возможностей и настроек.
>«1»	Конкретное значение разрешения в DPI, например, «300», «600», «1200» и т.д.

5.3. Системные принтеры

5.3.1. События

5.3.1.1. LoadFinished

Загрузка информации о принтерах завершена.

5.3.2. Функции

5.3.2.1. Load()

Вызывает асинхронную операцию для загрузки или обновления данных о доступных принтерах в системе. После вызова этой функции, информация о принтерах будет загружена в свойства [Информация о доступных принтерах](#) и [Информация о принтере по умолчанию](#).

Синтаксис

```
void Load();
```

Примеры



ПРИМЕР

Проверить и при необходимости загрузить информацию о принтерах, если она ранее не была загружена.

```
// Проверяем, была ли информация о принтерах уже загружена
if (!PrinterInfoCollection_1.Loaded) {
    // Если информация не была загружена, загружаем ее
    PrinterInfoCollection_1.Load();
}
```

5.3.3. Свойства

5.3.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

5.3.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиначный объект
>«1»	массив соответствующей размерности

5.3.3.3. Имена доступных принтеров

```
string PrinterNames
```

Список имен всех доступных принтеров в формате JSON.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовое поле список имен всех доступных принтеров.

```
Text_1.Text = PrinterInfoCollection_1.PrinterNames;
```

5.3.3.4. Имя принтера по умолчанию

```
string PrinterDefaultName
```

Имя принтера, установленного в системе как принтер по умолчанию.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовое поле имя принтера, установленного в системе как принтер по умолчанию.

```
Text_1.Text = PrinterInfoCollection_1.PrinterDefaultName;
```

5.3.3.5. Информация о доступных принтерах

```
string PrinterInfos
```

Детальная информация о всех принтерах, подключенных к системе, в формате JSON. Включает такие данные, как названия, описания, поддерживаемые разрешения печати и другие параметры.

Данные становятся доступными после выполнения метода **Load()**, который инициирует загрузку информации о принтерах.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовое поле детальную информацию о всех доступных принтерах.

```
PrinterInfoCollection_1.Load();  
Text_1.Text = PrinterInfoCollection_1.PrinterInfos;
```

5.3.3.6. Информация о принтере по умолчанию

```
string DefaultPrinterInfo
```

Детальная информация для принтера, установленного как принтер по умолчанию в системе, в формате JSON. Включает такие данные, как название, описание, поддерживаемые разрешения печати и другие параметры.

Данные становятся доступными после выполнения метода **Load()**, который инициирует загрузку информации о принтерах.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовое поле детальную информацию для принтера, установленного в системе как принтер по умолчанию.

```
PrinterInfoCollection_1.Load();  
Text_1.Text = PrinterInfoCollection_1.DefaultPrinterInfo;
```


5.3.3.7. Загружена ли информация о принтерах

bool Loaded

Была ли информация о принтерах загружена ранее с использованием метода [Load\(\)](#).



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

«true»	Метод Load() был вызван ранее и информация о принтерах уже загружена.
«false»	Метод Load() ещё не вызывался, или информация о принтерах устарела и требуется её обновление.

Примеры



ПРИМЕР

Проверить статус загрузки информации о принтерах и вывести результат в текстовое поле

```
Text_1.Text = PrinterInfoCollection_1.Loaded ? "Информация загружена" : "Требуется  
обновление данных";
```

6. Компоненты разметки

6.1. Линейная разметка

Позволяет линейно выравнивать объекты внутри контейнера (экранной формы или фигуры)

6.1.1. Свойства

6.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

6.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

6.1.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

6.1.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

6.1.1.5. Ориентация

Int4 Orientation

Направление выравнивания объектов внутри фигуры или формы.

Значение

1	ориентация по горизонтали
---	---------------------------

2	ориентация по вертикали
---	-------------------------

6.1.1.6. Промежуток

float Gap

Расстояние между объектами внутри фигуры или формы.

6.2. Плоская разметка

Позволяет расположить внутри фигуры или формы объекты с разной ориентацией. Ориентация объектов зависит от размеров и положения фигуры или формы. Промежутки между объектами невозможно изменить.

6.2.1. Свойства

6.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

6.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

6.3. Вес элемента разметки

Позволяет задавать объектам разметки приоритет размера по отношению к другим объектам фигуры или формы. Компонент работает как дополнение компонентов **Плоская разметка** и **Линейная разметка**.

6.3.1. Свойства

6.3.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

6.3.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

6.3.1.3. Значимость

```
float Weight
```

Вес элемента разметки.

6.4. Ограничение размеров элемента разметки

6.4.1. Свойства

6.4.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

6.4.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

6.4.1.3. Минимальная ширина

```
double MinWidth
```

Минимально допустимая ширина элемента.

6.4.1.4. Минимальная высота

```
double MinHeight
```

Минимально допустимая высота элемента.

6.4.1.5. Максимальная ширина

```
double MaxWidth
```

Максимально допустимая ширина элемента

6.4.1.6. Максимальная высота

```
double MaxHeight
```

Максимально допустимая высота элемента

6.5. Выравнивание элемента разметки

6.5.1. Свойства

6.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

6.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

6.5.1.3. Выравнивание по горизонтали

```
int4 HorizontalAlignment
```

Расположение элемента внутри его контейнера по горизонтальной оси.

Значение

«Слева»	Элемент будет расположен в самой левой части контейнера.
«По центру»	Элемент будет расположен посередине контейнера, обеспечивая равное расстояние от левой и правой границ.
«Справа»	Элемент будет расположен в самой правой части контейнера.

Примеры



ПРИМЕР

Установить горизонтальное выравнивание кнопок внутри контейнера «Rectangle_1»

```
Rectangle_1.Button_1.LayoutAlignment_1.HorizontalAlignment = 0; // Выравнивание по левому краю  
Rectangle_1.Button_2.LayoutAlignment_1.HorizontalAlignment = 1; // Выравнивание по центру  
Rectangle_1.Button_3.LayoutAlignment_1.HorizontalAlignment = 2; // Выравнивание по правому краю
```

6.5.1.4. Выравнивание по вертикали

```
int4 VerticalAlignment
```

Расположение элемента внутри его контейнера по вертикальной оси.

Значение

«Сверху»	Элемент будет располагаться у верхнего края контейнера.
«По центру»	Элемент будет расположен посередине контейнера, обеспечивая равное расстояние от верхнего и нижнего краёв.
«Снизу»	Элемент будет располагаться у нижнего края контейнера.

Примеры



ПРИМЕР

Установить вертикальное выравнивание кнопок внутри контейнера «Rectangle_1»

```
Rectangle_1.Button_1.LayoutAlignment_1.VerticalAlignment = 0; // Выравнивание по
верхнему краю
Rectangle_1.Button_2.LayoutAlignment_1.VerticalAlignment = 1; // Выравнивание по
центру
Rectangle_1.Button_3.LayoutAlignment_1.VerticalAlignment = 2; // Выравнивание по
нижнему краю
```

6.6. Отступы элемента разметки

6.6.1. Свойства

6.6.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

6.6.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

6.6.1.3. Сверху

double Top

Отступ элемента от верхней границы контейнера.

6.6.1.4. Снизу

double Bottom

Отступ элемента от нижней границы контейнера.

6.6.1.5. Слева

```
double Left
```

Отступ элемента от левой границы контейнера.

6.6.1.6. Справа

```
double Right
```

Отступ элемента от правой границы контейнера.

7. Графические компоненты

7.1. Линейный градиент

7.1.1. Свойства

7.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

7.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

7.1.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

7.1.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

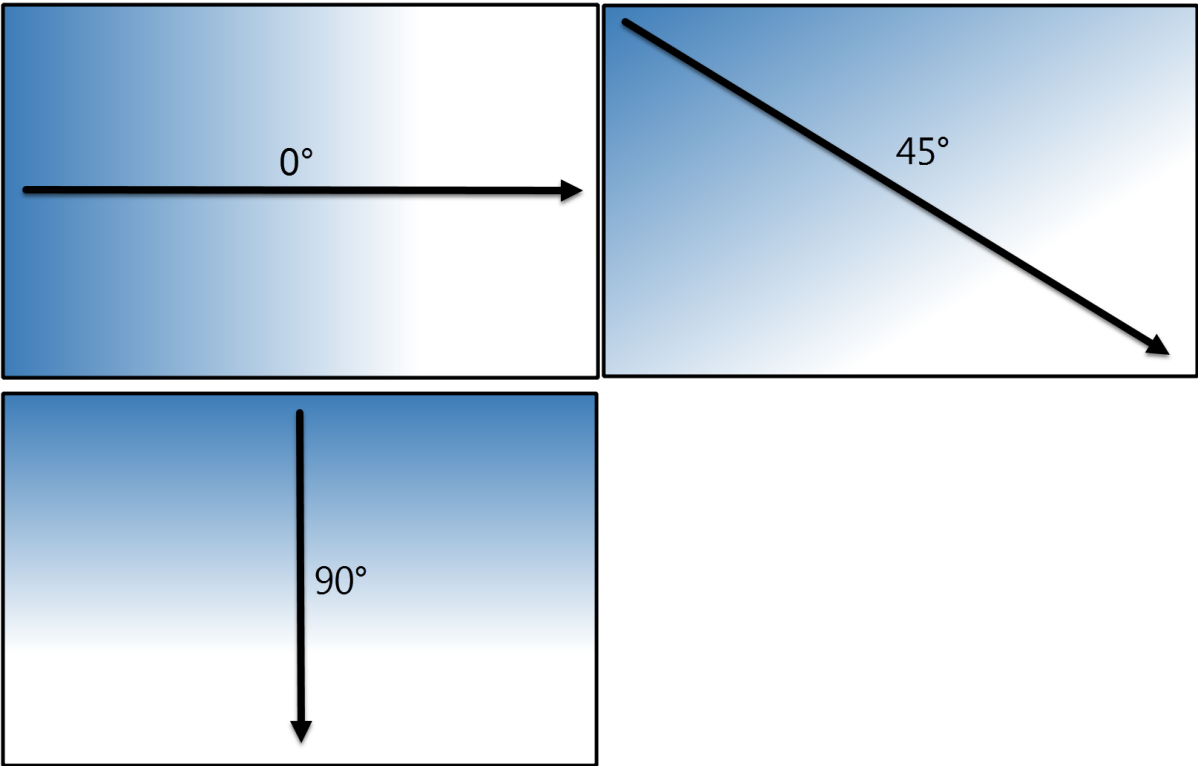
В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

7.1.1.5. Угол поворота

Float Rotation

Угол наклона градиентной линии, которая задает направление градиента. 0 градусов соответствует линейному переходу слева направо. С увеличением градуса линия градиента поворачивается по часовой стрелке.



7.1.1.6. Видимость

bool Visible

Видимость линейного градиента на графическом объекте.

Значение

«true»	Линейный градиент видим и отображается на графическом объекте.
«false»	Линейный градиент становится скрытым и не будет отображаться на графическом объекте.

7.1.1.7. Непрозрачность

double Opacity

Степень прозрачности линейного градиента на графическом объекте. Измеряется в диапазоне от «0.0» до «1.0», где

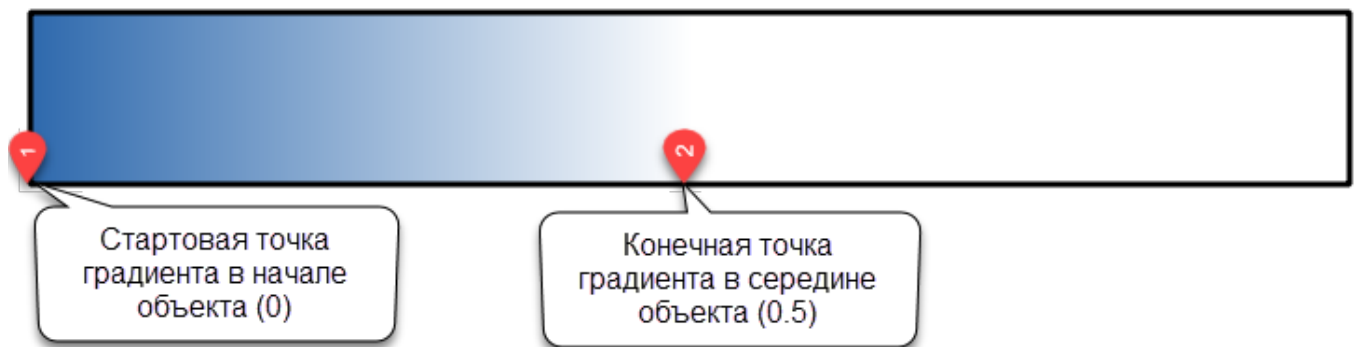
- «0.0» - градиент полностью прозрачный;
- «1.0» - градиент полностью непрозрачный.

Свойства точек градиента:

7.1.1.8. Положение

Float Position

Положение точек начала или конца цветового перехода. Измеряется от «0» до «1», где «0» - начало объекта, «1» - конец объекта. На картинке и в примере ниже показано, что старт линейного градиента находится в самом начале объекта (значение «0»), а конец градиента наступает уже к середине объекта (значение «0.5»).



Примеры



ПРИМЕР

Задать стартовую позицию градиента в начале объекта и конечную позицию градиента в середине объекта.

```
Object.Gradient.GradientPoint_1.Position = 0;  
Object.Gradient.GradientPoint_2.Position = 0.5;
```

7.1.1.9. Цвет

uInt4 Color

Цвет в начальной или конечной точке градиента. Выбирается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Установить начальный цвет градиента - черный и конечный цвет градиента - белый.

```
Object.Gradient.GradientPoint_1.Color = 0xff000000;  
Object.Gradient.GradientPoint_2.Color = 0xffffffff;
```

7.2. Радиальный градиент

7.2.1. Свойства

7.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

7.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

7.2.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

7.2.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

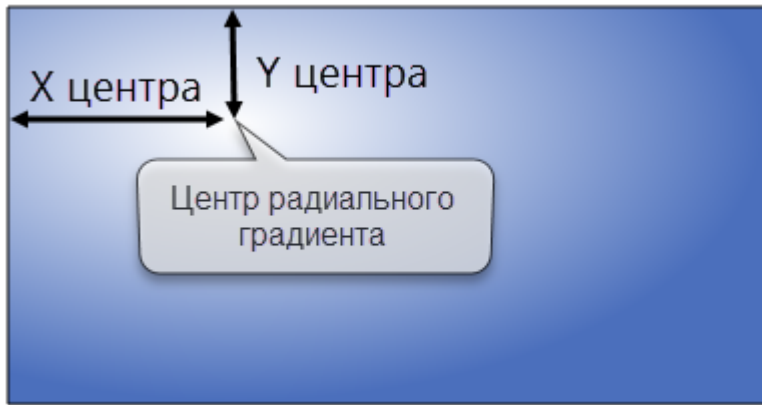
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

7.2.1.5. X центра и Y центра

Float XCenter

Float YCenter

Координаты центра радиального градиента по горизонтали и вертикали.



7.2.1.6. Видимость

bool Visible

Видимость радиального градиента на графическом объекте.

Значение

«true»	Радиальный градиент видим и отображается на графическом объекте.
«false»	Радиальный градиент становится скрытым и не будет отображаться на графическом объекте.

7.2.1.7. Непрозрачность

double Opacity

Степень прозрачности радиального градиента на графическом объекте. Измеряется в диапазоне от «0.0» до «1.0», где

- «0.0» - градиент полностью прозрачный;
- «1.0» - градиент полностью непрозрачный.

7.2.1.8. Размер

uint1 Size

Радиус распространения радиального градиента на графическом объекте.

Значение по умолчанию «По дальнему углу».

Значение

«По дальнему углу»	
«По дальней стороне»	
«По ближайшему углу»	
«По ближайшей стороне»	

Свойства точек градиента:

7.2.1.9. Положение

Float Position

Положение точек начала или конца цветового перехода. Измеряется от «0» до «1», где «0» - начало объекта, «1» - конец объекта.

7.2.1.10. Цвет

uInt4 Color

Цвет в начальной или конечной точке градиента. Выбирается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

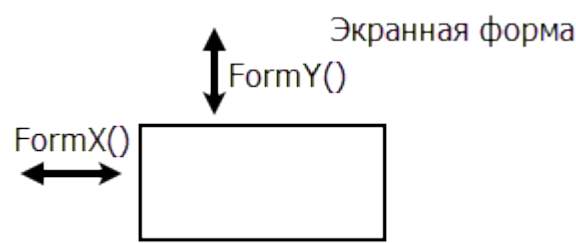
7.3. Соединительная линия

Компонент **Соединительная линия** используется для создания и визуализации участков трубопровода. Компонент обеспечивает возможность создания сложных участков трубопровода из одной линии путем перемещения отдельных отрезков линии.

7.3.1. Функции

7.3.1.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры

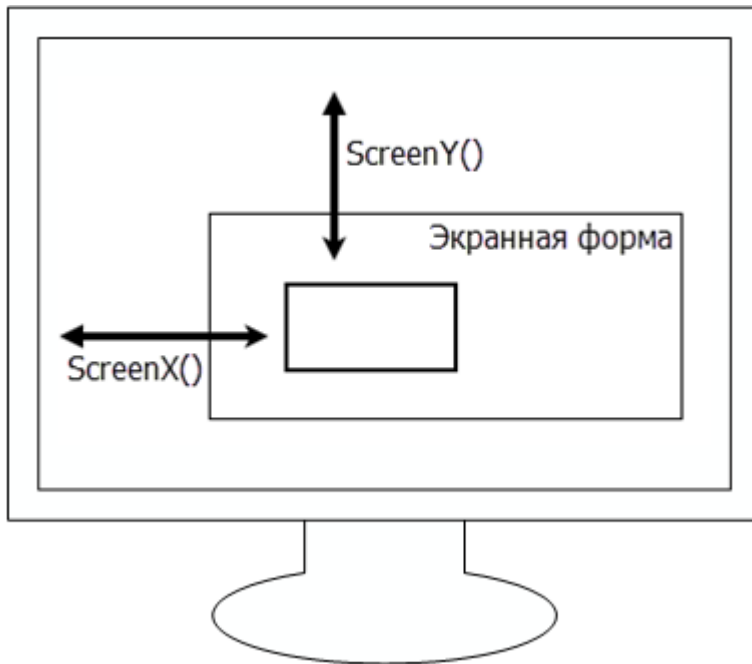
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

7.3.1.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

7.3.2. Свойства

7.3.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

7.3.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

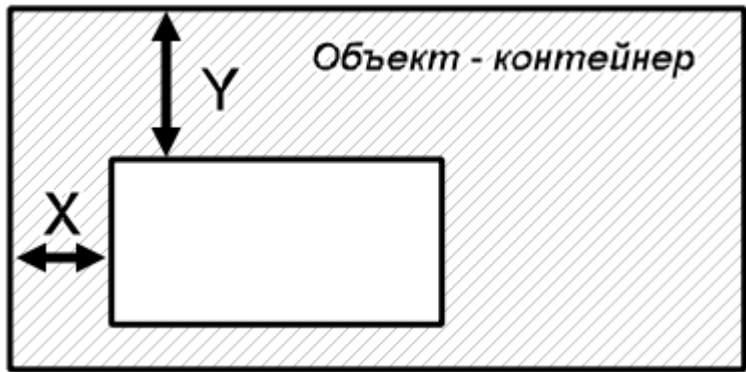
«1»	одиночный объект
>«1»	массив соответствующей размерности

7.3.2.3. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

7.3.2.4. Z-значение

Float ZValue

Порядок отображения объекта на экране. Чем выше значение свойства **Z-значение**, тем объект отображается выше, перекрывая объекты с меньшими значениями свойства **Z-значение**.

7.3.2.5. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

7.3.2.6. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

7.3.2.7. Отражение

uint1 Flip

Отражение объекта относительно горизонтальной и вертикальной осей.

Значение

«Без отражения »	Объект отображается без изменений.
«По горизонтали»	Объект отражается по горизонтали, создавая зеркальное отображение относительно вертикальной оси.
«По вертикали»	Объект отражается по вертикали, создавая зеркальное отображение относительно горизонтальной оси
«По горизонтали и вертикали»	Объект отражается одновременно и по горизонтали, и по вертикали, создавая зеркальное отображение относительно обеих осей.

7.3.2.8. Видимость

bool Visible

Видимость объекта на форме.

Значение

true	Объект виден.
false	Объект скрыт.

7.3.2.9. Включено

bool Enabled

Активность объекта.



Значение

true	Объект активен.
false	Объект заблокирован.

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

7.3.2.10. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	Фокус на объекте.
false	Фокус вне объекта.

Примеры



ПРИМЕР

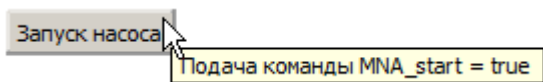
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

7.3.2.11. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

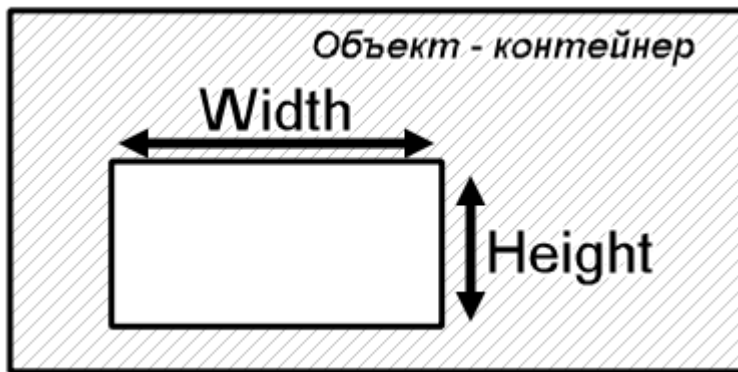
```
Object.Tooltip = "Подача команды MNA_start = true";
```

7.3.2.12. Ширина и Высота

float Width

float Height

Ширина и высота объекта в пикселях.



7.3.2.13. Декоратор

```
bool EnableDecoration
```

Изменение внешнего вида соединительной линии для создания сходства с трубопроводом.

Значение

true	Линия становится похожей на трубопровод.
false	Оставляет линию стандартного вида.

7.3.2.14. Цвет пера

```
UInt4 PenColor
```

Цвет внешней границы объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.PenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.



```
Object.PenColor = 4278190080;
```

7.3.2.15. Стил ь пера

UInt2 PenStyle

Стил ь внешней границы объекта.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет пера»	«0»	
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стил ь границы (короткий пунктир) для объекта.

```
Object.PenStyle = 3;
```

7.3.2.16. Толщина пера

Float PenWidth

Толщина внешней границы объекта. Измеряется в пикселях.

Примеры



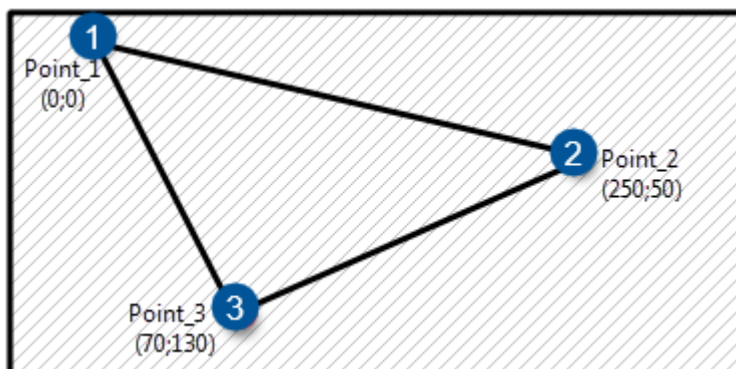
ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.PenWidth = 5;
```

Свойства точек

Каждая точка, в составе многоугольника имеет координаты: свойство **X** и свойство **Y**. Началом отсчета координат (точка - 0;0) считается первая точка многоугольника. Координаты остальных точек отсчитываются относительно первой точки.



ПРИМЕР

Изменить координаты точки 2 относительно координат точки 1.

```
Figure.Point_2.X = Figure.Point_1.X + 250;  
Figure.Point_2.Y = Figure.Point_1.Y + 50;
```

7.3.3. События

7.3.3.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```

ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.3.3.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.3.3.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.3.3.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```

ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.3.3.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.3.3.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.3.3.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```


7.3.3.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.3.3.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события

PointX	Float	функция, получающая координату точки нажатия по оси X <div><pre>float8 PointX(uint4)</pre></div> параметр uint4 - порядковый номер нажатия
PointY	Float	функция, получающая координату точки нажатия по оси Y <div><pre>float8 PointY(uint4)</pre></div> параметр uint4 - порядковый номер нажатия

Примеры

ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name",Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o",Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp",Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```

ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

7.3.3.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события

PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры



ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «Ctrl» увеличивать масштаб объекта. В обработчике события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

7.3.3.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки поднятия пальца (пера) по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер поднятия пальца (пера)</p>
PointY	Float	<p>функция, получающая координату точки поднятия пальца (пера) по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер поднятия пальца (пера)</p>

Примеры



ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу OpenSW значение «1». В обработчике события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;
Ellipse_1.Y = 341;
SW_1.OpenSW = 1;
```

7.3.3.12. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

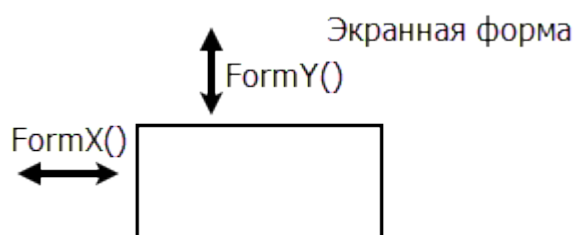
7.4. Место соединения

Компонент для управления местом соединений при создании схем. Чтобы выбрать тип и цвет места соединения, используйте свойства элемента **Место соединения**.

7.4.1. Функции

7.4.1.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



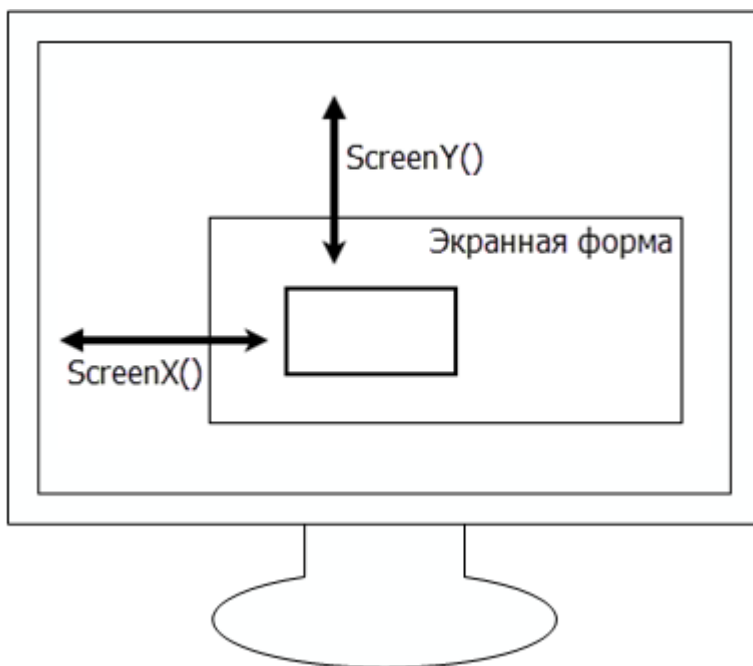
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

7.4.1.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

7.4.2. Свойства

7.4.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

7.4.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

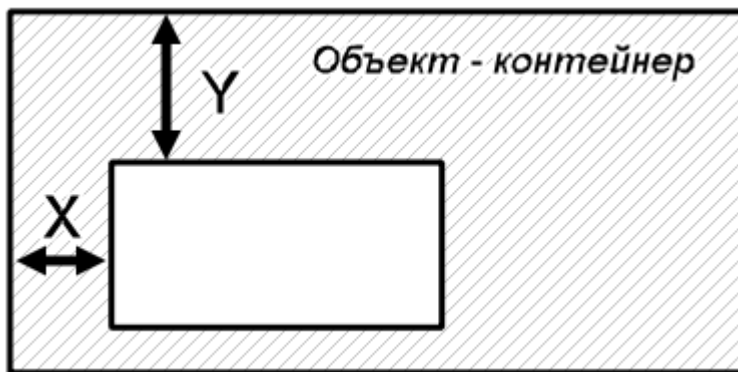
«1»	одиночный объект
>«1»	массив соответствующей размерности

7.4.2.3. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

7.4.2.4. Z-значение

Float ZValue

Порядок отображения объекта на экране. Чем выше значение свойства **Z-значение**, тем объект отображается выше, перекрывая объекты с меньшими значениями свойства **Z-значение**.

7.4.2.5. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

7.4.2.6. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

7.4.2.7. Отражение

uint1 Flip

Отражение объекта относительно горизонтальной и вертикальной осей.

Значение

«Без отражения »	Объект отображается без изменений.
«По горизонтали»	Объект отражается по горизонтали, создавая зеркальное отображение относительно вертикальной оси.
«По вертикали»	Объект отражается по вертикали, создавая зеркальное отображение относительно горизонтальной оси
«По горизонтали и вертикали»	Объект отражается одновременно и по горизонтали, и по вертикали, создавая зеркальное отображение относительно обеих осей.

7.4.2.8. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	Объект виден.
false	Объект скрыт.

7.4.2.9. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	Объект активен.
false	Объект заблокирован.

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

7.4.2.10. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	Фокус на объекте.
false	Фокус вне объекта.

Примеры



ПРИМЕР

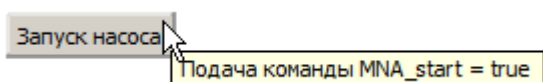
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

7.4.2.11. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

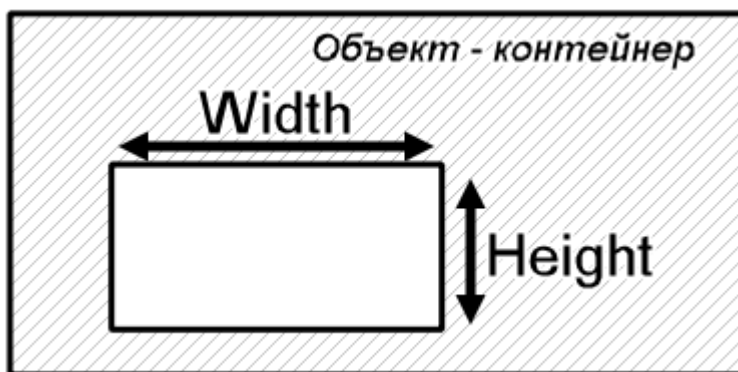
```
Object.Tooltip = "Подача команды MNA_start = true";
```

7.4.2.12. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



7.4.2.13. Декоратор

```
bool EnableDecoration
```

Декоративный внешний вид соединителей (мест соединений).

Значение

true	Соединители приобретают фигурный вид.
false	Оставляет соединители в стандартном визуальном состоянии.

7.4.2.14. Тип соединения

```
uint4 JointType
```

Тип соединителей (мест соединений).

Перед выбором типа соединения убедитесь, что свойство **Декоратор** установлено в «true», чтобы увидеть различные варианты типов соединений.

Значение

0	
1	
2	
3	
4	

7.4.2.15. Цвет пера

UInt4 PenColor

Цвет внешней границы объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.PenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.






```
Object.PenColor = 4278190080;
```

7.4.2.16. Стиль пера

UInt2 PenStyle

Стиль внешней границы объекта.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет пера»	«0»	
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта.

```
Object.PenStyle = 3;
```

7.4.2.17. Толщина пера

Float PenWidth

Толщина внешней границы объекта. Измеряется в пикселях.

Примеры

ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.PenWidth = 5;
```

7.4.2.18. Цвет заливки

UInt4 BrushColor

Цвет внутренней заливки объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры

ПРИМЕР

Залить объект красным цветом (цвет в шестнадцатеричной системе счисления).

```
Object.BrushColor = 0xffff0000;
```

ПРИМЕР

Залить объект красным цветом (цвет в десятичной системе счисления).

```
Object.BrushColor = 4294901760;
```



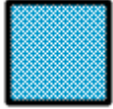

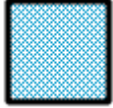
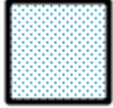


7.4.2.19. Стиль заливки

UInt2 BrushStyle

Заливает объект одним из стилей заливки.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	

Значение в дизайнере	Значение в коде	Вид границы
«Точечная 1»	«2»	
«Точечная 2»	«3»	
«Точечная 3»	«4»	
«Точечная 4»	«5»	
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



ПРИМЕР

Залить объект сплошным цветом.

```
Object.BrushStyle = 1;
```

7.4.3. События

7.4.3.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.4.3.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.4.3.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.4.3.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.4.3.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.4.3.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.4.3.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```

**ПРИМЕР**

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

**ПРИМЕР**

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.4.3.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

**ПРИМЕР**

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```

**ПРИМЕР**

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.4.3.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки нажатия по оси X <div>float8 PointX(uint4)</div> <div>параметр uint4 - порядковый номер нажатия</div>
PointY	Float	функция, получающая координату точки нажатия по оси Y <div>float8 PointY(uint4)</div> <div>параметр uint4 - порядковый номер нажатия</div>

Примеры



ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name",Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o",Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp",Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```



ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

7.4.3.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки нажатия по оси X <pre>float8 PointX(uint4)</pre> параметр uint4 - порядковый номер нажатия
PointY	Float	функция, получающая координату точки нажатия по оси Y <pre>float8 PointY(uint4)</pre> параметр uint4 - порядковый номер нажатия

Примеры



ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «Ctrl» увеличивать масштаб объекта. В обработчике события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

7.4.3.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки поднятия пальца (пера) по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер поднятия пальца (пера)</p>
PointY	Float	<p>функция, получающая координату точки поднятия пальца (пера) по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер поднятия пальца (пера)</p>

Примеры



ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу OpenSW значение «1». В обработчике события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;
Ellipse_1.Y = 341;
SW_1.OpenSW = 1;
```

7.4.3.12. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

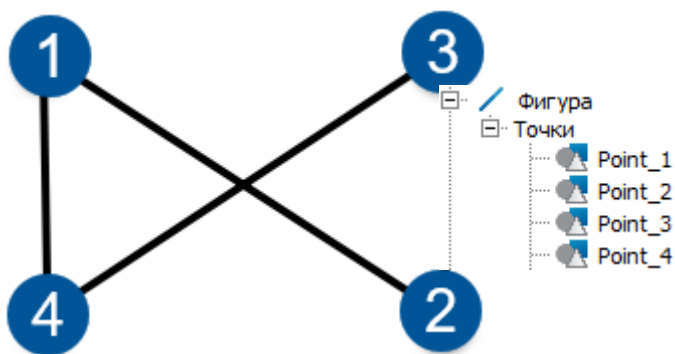
7.5. Линия

Элемент для создания следующих типов объектов:

- прямые линии;
- ломаные линии;
- многоугольники.



Каждый объект, созданный с помощью элемента **Линия** представлен набором точек, связь которых образует объект.



7.5.1. События

7.5.1.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```

**ПРИМЕР**

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

**ПРИМЕР**

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.5.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

**ПРИМЕР**

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```

**ПРИМЕР**

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

**ПРИМЕР**

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.5.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.5.1.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```

ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.5.1.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.5.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.5.1.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.5.1.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие

Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.5.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры

ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name",Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o",Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp",Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```

ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

7.5.1.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<div>функция, получающая координату точки нажатия по оси X</div> <div><pre>float8 PointX(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>
PointY	Float	<div>функция, получающая координату точки нажатия по оси Y</div> <div><pre>float8 PointY(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>

Примеры



ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «**Ctrl**» увеличивать масштаб объекта. В обработке события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

7.5.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов (« Ctrl », « Shift », « Alt »), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки поднятия пальца (пера) по оси X <div style="background-color: #e6f2ff; padding: 5px; margin: 5px 0;">float8 PointX(uint4)</div> параметр uint4 - порядковый номер поднятия пальца (пера)
PointY	Float	функция, получающая координату точки поднятия пальца (пера) по оси Y <div style="background-color: #e6f2ff; padding: 5px; margin: 5px 0;">float8 PointY(uint4)</div> параметр uint4 - порядковый номер поднятия пальца (пера)

Примеры



ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу *OpenSW* значение «1». В обработке события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;
Ellipse_1.Y = 341;
SW_1.OpenSW = 1;
```

7.5.1.12. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

7.5.2. Свойства

7.5.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

7.5.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

7.5.2.3. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

7.5.2.4. Z-значение

Float ZValue

Порядок отображения объекта на экране. Чем выше значение свойства **Z-значение**, тем объект отображается выше, перекрывая объекты с меньшими значениями свойства **Z-значение**.

7.5.2.5. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

7.5.2.6. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

7.5.2.7. Отражение

uint1 Flip

Отражение объекта относительно горизонтальной и вертикальной осей.

Значение

«Без отражения »	Объект отображается без изменений.
«По горизонтали»	Объект отражается по горизонтали, создавая зеркальное отображение относительно вертикальной оси.
«По вертикали»	Объект отражается по вертикали, создавая зеркальное отображение относительно горизонтальной оси
«По горизонтали и вертикали»	Объект отражается одновременно и по горизонтали, и по вертикали, создавая зеркальное отображение относительно обеих осей.

7.5.2.8. Видимость

bool Visible

Видимость объекта на форме.

Значение

true	Объект виден.
false	Объект скрыт.

7.5.2.9. Включено

bool Enabled

Активность объекта.



Значение

true	Объект активен.
false	Объект заблокирован.

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

7.5.2.10. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	Фокус на объекте.
false	Фокус вне объекта.

Примеры



ПРИМЕР

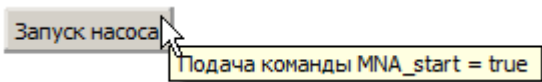
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

7.5.2.11. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

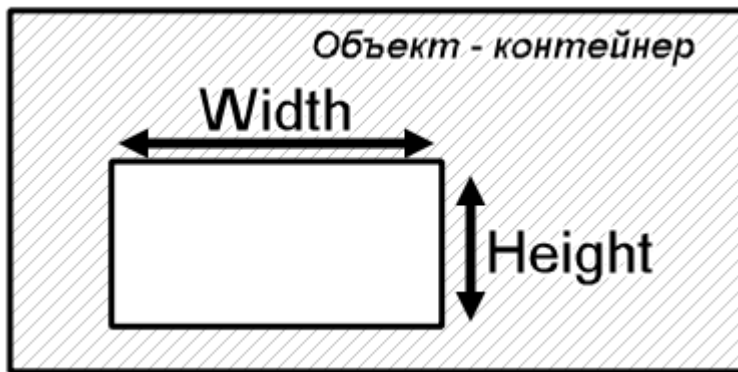
```
Object.Tooltip = "Подача команды MNA_start = true";
```

7.5.2.12. Ширина и Высота

float Width

float Height

Ширина и высота объекта в пикселях.



7.5.2.13. Цвет пера

UInt4 PenColor

Цвет внешней границы объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.PenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.

```
Object.PenColor = 4278190080;
```

7.5.2.14. Стиль пера

UInt2 PenStyle


Стиль внешней границы объекта.

Значение

Значение в дизайнера	Значение в коде	Вид границы
«Нет пера»	«0»	

Значение в дизайнере	Значение в коде	Вид границы
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта.


```
Object.PenStyle = 3;
```

7.5.2.15. Толщина пера

Float PenWidth

Толщина внешней границы объекта. Измеряется в пикселях.

Примеры



ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.PenWidth = 5;
```


7.5.2.16. Цвет заливки

UInt4 BrushColor

Цвет внутренней заливки объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Залить объект красным цветом (цвет в шестнадцатеричной системе счисления).

```
Object.BrushColor = 0xffff0000;
```



ПРИМЕР

Залить объект красным цветом (цвет в десятичной системе счисления).

```
Object.BrushColor = 4294901760;
```





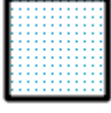

7.5.2.17. Стиль заливки

UInt2 BrushStyle


Заливает объект одним из стилей заливки.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	
«Точечная 1»	«2»	
«Точечная 2»	«3»	

Значение в дизайнере	Значение в коде	Вид границы
«Точечная 3»	«4»	
«Точечная 4»	«5»	
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



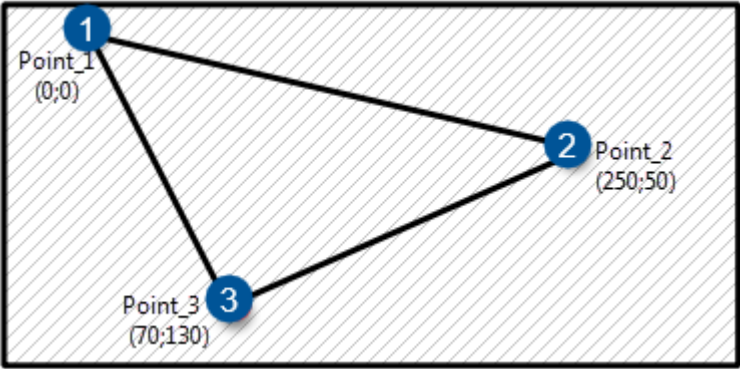
ПРИМЕР

Залить объект сплошным цветом.

```
Object.BrushStyle = 1;
```

Свойства точек

Каждая точка, в составе многоугольника имеет координаты: свойство X и свойство Y. Началом отсчета координат (точка - 0;0) считается первая точка многоугольника. Координаты остальных точек отсчитываются относительно первой точки.



ПРИМЕР

Изменить координаты точки 2 относительно координат точки 1.

```
Figure.Point_2.X = Figure.Point_1.X + 250;  
Figure.Point_2.Y = Figure.Point_1.Y + 50;
```

7.6. Прямоугольник

7.6.1. Свойства

7.6.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

7.6.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

7.6.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

7.6.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

7.6.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



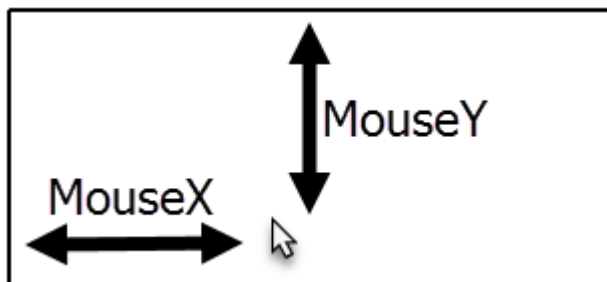
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

7.6.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

7.6.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```

**ПРИМЕР**

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

7.6.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры

**ПРИМЕР**

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```

**ПРИМЕР**

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```

**ПРИМЕР**

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

7.6.1.9. Видимость

bool Visible

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

7.6.1.10. Включено

bool Enabled

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

7.6.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

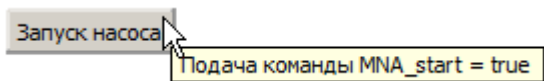
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

7.6.1.12. Всплывающая подсказка

```
string Tooltip
```

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

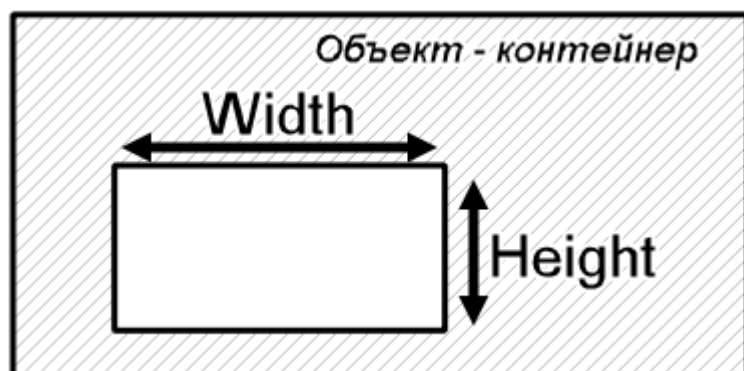
```
Object.Tooltip = "Подача команды MNA_start = true";
```

7.6.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

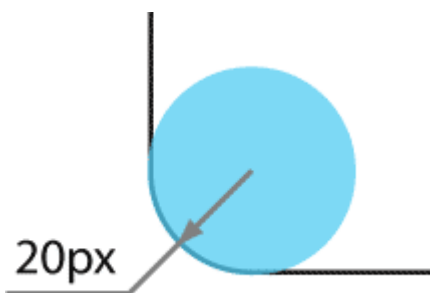
Ширина и высота объекта в пикселях.



7.6.1.14. Радиус скругления

Float RoundingRadius

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



7.6.1.15. Цвет пера

UInt4 PenColor

Цвет внешней границы объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.PenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.

```
Object.PenColor = 4278190080;
```

7.6.1.16. Стил ь пера

UInt2 PenStyle

Стил ь внешней границы объекта.

Значение

Значение в дизай нере	Значение в коде	Вид границы
«Нет пера»	«0»	
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта.

```
Object.PenStyle = 3;
```

7.6.1.17. Толщина пера

Float PenWidth

Толщина внешней границы объекта. Измеряется в пикселях.

Примеры



ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.PenWidth = 5;
```

7.6.1.18. Цвет заливки

UInt4 BrushColor

Цвет внутренней заливки объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Залить объект красным цветом (цвет в шестнадцатеричной системе счисления).

```
Object.BrushColor = 0xffff0000;
```



ПРИМЕР

Залить объект красным цветом (цвет в десятичной системе счисления).




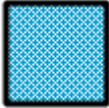

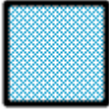



```
Object.BrushColor = 4294901760;
```

7.6.1.19. Стил ь заливки

UInt2 BrushStyle

Заливает объект одним из стилей заливки.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	
«Точечная 1»	«2»	
«Точечная 2»	«3»	
«Точечная 3»	«4»	
«Точечная 4»	«5»	
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



ПРИМЕР

Залить объект сплошным цветом.

```
Object.BrushStyle = 1;
```

7.6.2. События

7.6.2.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.6.2.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.6.2.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.6.2.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```




ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.6.2.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.6.2.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.6.2.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.6.2.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

PRIMER

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```

PRIMER

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

PRIMER

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.6.2.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<div>функция, получающая координату точки нажатия по оси X</div> <div><pre>float8 PointX(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>
PointY	Float	<div>функция, получающая координату точки нажатия по оси Y</div> <div><pre>float8 PointY(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>

Примеры



ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name", Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o", Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp", Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```



ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

7.6.2.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки нажатия по оси X <pre>float8 PointX(uint4)</pre> параметр uint4 - порядковый номер нажатия
PointY	Float	функция, получающая координату точки нажатия по оси Y <pre>float8 PointY(uint4)</pre> параметр uint4 - порядковый номер нажатия

Примеры

ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «Ctrl» увеличивать масштаб объекта. В обработчике события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

7.6.2.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки поднятия пальца (пера) по оси X <div><pre>float8 PointX(uint4)</pre>параметр uint4 - порядковый номер поднятия пальца (пера)</div>
PointY	Float	функция, получающая координату точки поднятия пальца (пера) по оси Y <div><pre>float8 PointY(uint4)</pre>параметр uint4 - порядковый номер поднятия пальца (пера)</div>

Примеры

ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу OpenSW значение «1». В обработчике события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;
Ellipse_1.Y = 341;
SW_1.OpenSW = 1;
```

7.6.2.12. FocusChanged

Изменение фокуса у объекта.

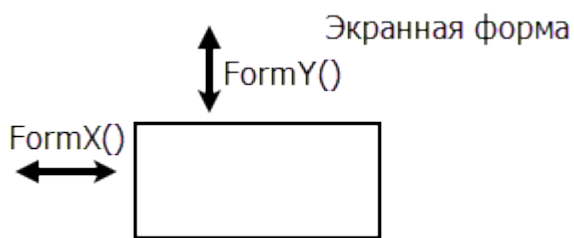
Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

7.6.3. Функции

7.6.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



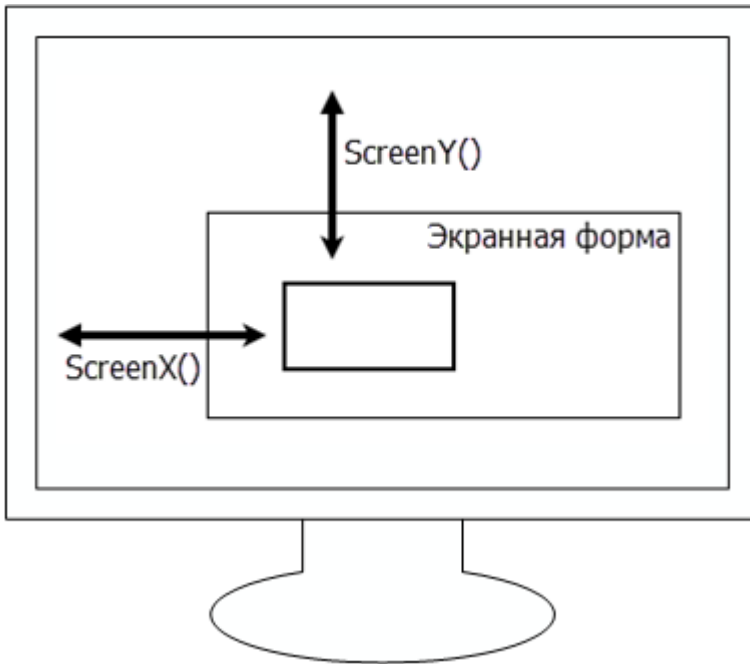
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

7.6.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```


7.7. Эллипс

7.7.1. Свойства

7.7.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

7.7.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

7.7.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

7.7.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

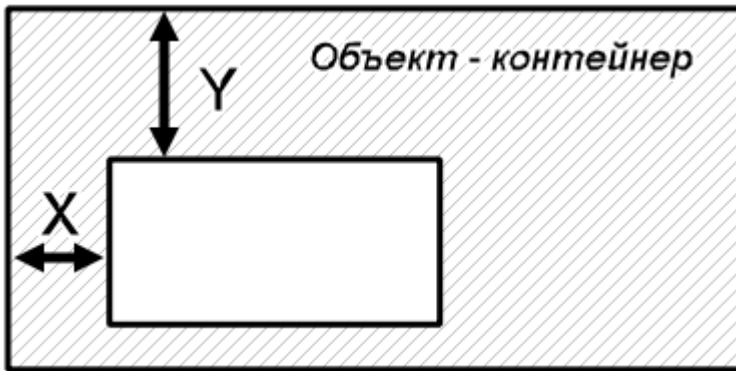
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

7.7.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



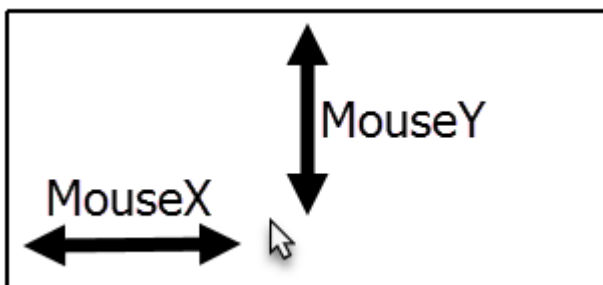
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

7.7.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

7.7.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

7.7.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

7.7.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

7.7.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

7.7.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

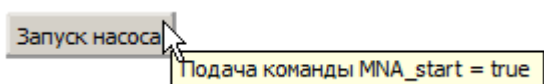
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

7.7.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

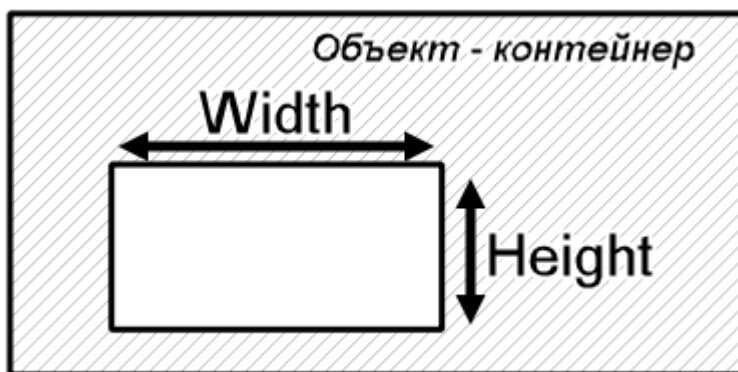
```
Object.Tooltip = "Подача команды MNA_start = true";
```

7.7.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



7.7.1.14. Цвет пера

```
UInt4 PenColor
```

Цвет внешней границы объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.PenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.

```
Object.PenColor = 4278190080;
```

7.7.1.15. Стиль пера

UInt2 PenStyle

Стиль внешней границы объекта.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет пера»	«0»	
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта.

```
Object.PenStyle = 3;
```

7.7.1.16. Толщина пера

Float PenWidth

Толщина внешней границы объекта. Измеряется в пикселях.

Примеры



ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.PenWidth = 5;
```

7.7.1.17. Цвет заливки

UInt4 BrushColor

Цвет внутренней заливки объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Залить объект красным цветом (цвет в шестнадцатеричной системе счисления).

```
Object.BrushColor = 0xffff0000;
```



ПРИМЕР

Залить объект красным цветом (цвет в десятичной системе счисления).




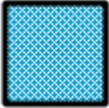



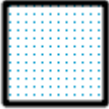

```
Object.BrushColor = 4294901760;
```

7.7.1.18. Стиль заливки

UInt2 BrushStyle

Заливает объект одним из стилей заливки.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	
«Точечная 1»	«2»	
«Точечная 2»	«3»	
«Точечная 3»	«4»	
«Точечная 4»	«5»	
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



ПРИМЕР

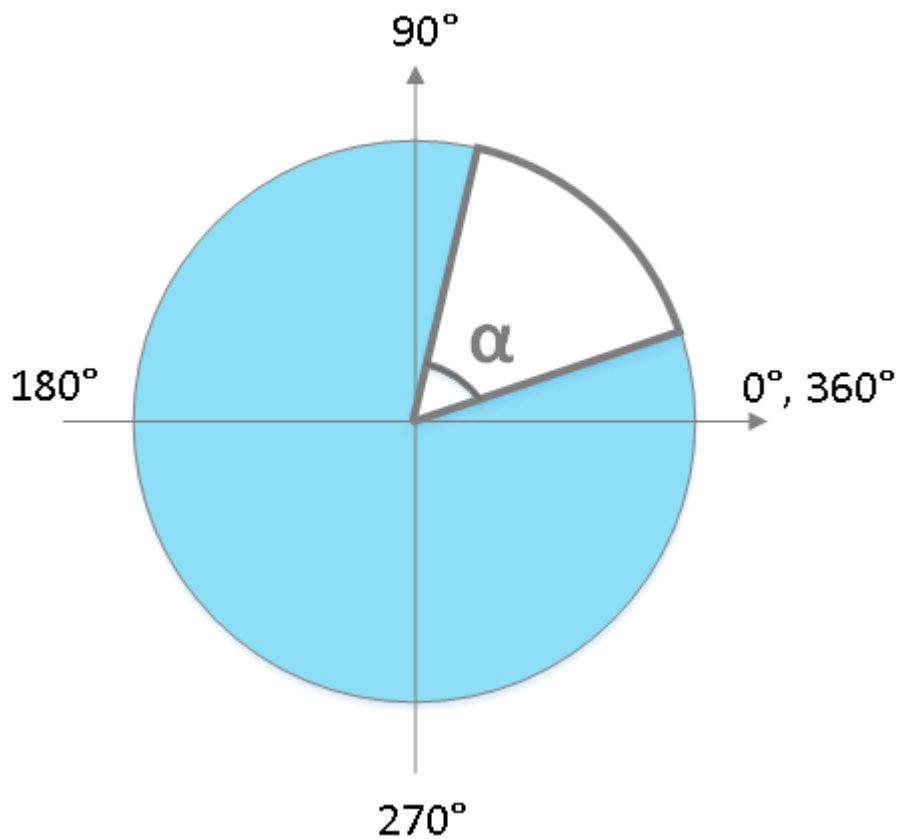
Залить объект сплошным цветом.

```
Object.BrushStyle = 1;
```

7.7.1.19. Размер сектора

Float PieAngle

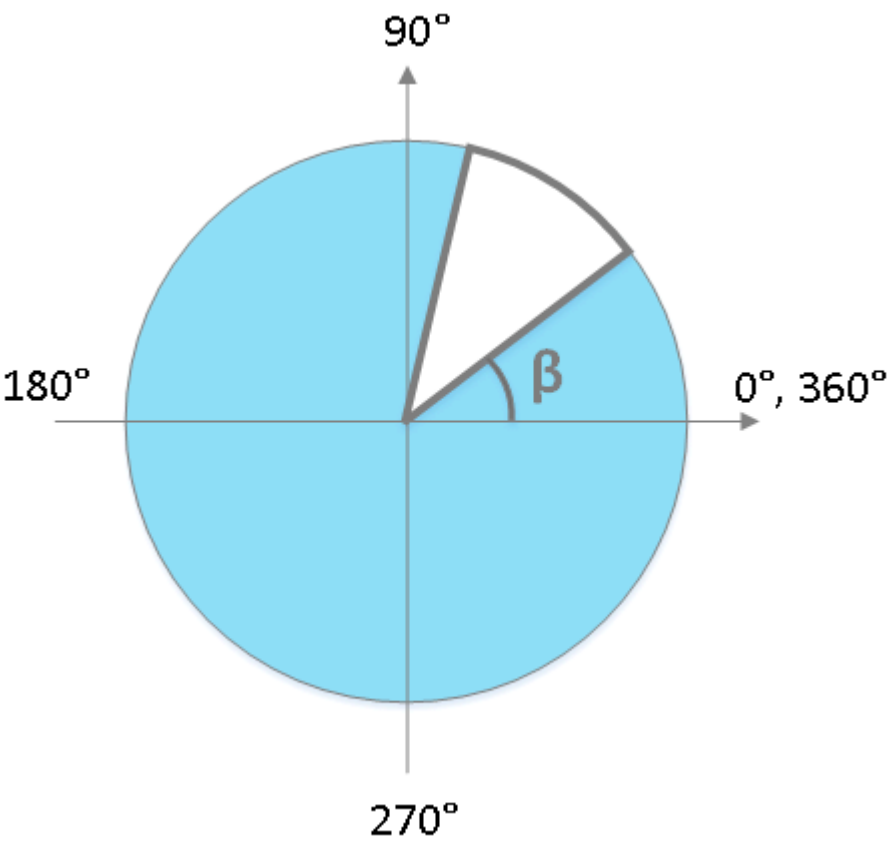
Угол отображаемого сектора эллипса. Измеряется в градусах.



7.7.1.20. Начало сектора

Float PieAngleStart

Угол, который считается точкой отсчета при очерчивании сектора эллипса. Измеряется в градусах.



7.7.2. События

7.7.2.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.7.2.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.7.2.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.7.2.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.7.2.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.7.2.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```


7.7.2.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.7.2.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие

Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```

ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.7.2.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки нажатия по оси X <div>float8 PointX(uint4)</div> <div>параметр uint4 - порядковый номер нажатия</div>
PointY	Float	функция, получающая координату точки нажатия по оси Y <div>float8 PointY(uint4)</div> <div>параметр uint4 - порядковый номер нажатия</div>

Примеры



ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name", Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o", Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp", Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```



ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

7.7.2.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры

ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «**Ctrl**» увеличивать масштаб объекта. В обработчике события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

7.7.2.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов (« Ctrl », « Shift », « Alt »), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки поднятия пальца (пера) по оси X <div><pre>float8 PointX(uint4)</pre>параметр uint4 - порядковый номер поднятия пальца (пера)</div>
PointY	Float	функция, получающая координату точки поднятия пальца (пера) по оси Y <div><pre>float8 PointY(uint4)</pre>параметр uint4 - порядковый номер поднятия пальца (пера)</div>

Примеры

ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу OpenSW значение «1». В обработчике события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;
Ellipse_1.Y = 341;
SW_1.OpenSW = 1;
```

7.7.2.12. FocusChanged

Изменение фокуса у объекта.

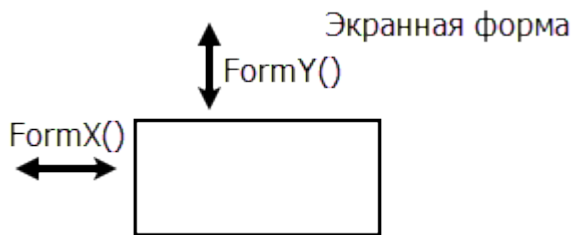
Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

7.7.3. Функции

7.7.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



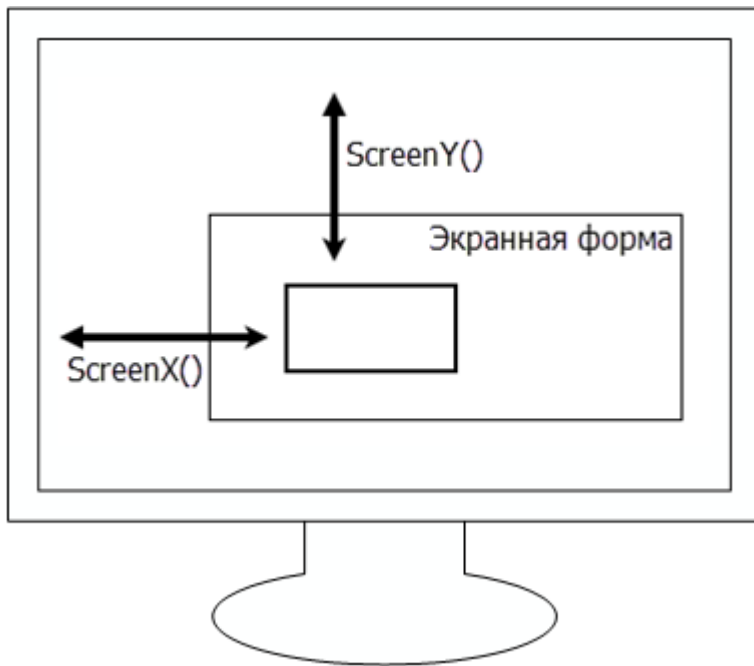
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

7.7.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

7.8. Изображение

Компонент для вставки растрового изображения из файла.

7.8.1. Свойства

7.8.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

7.8.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

7.8.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

7.8.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

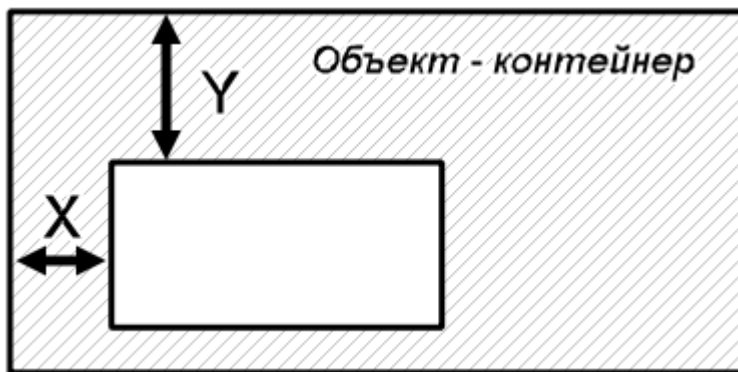
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

7.8.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



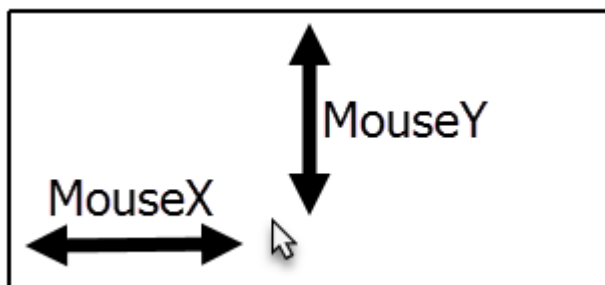
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

7.8.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

7.8.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

7.8.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

7.8.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

7.8.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

7.8.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

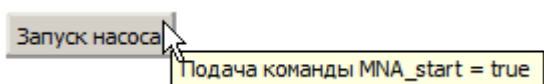
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

7.8.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

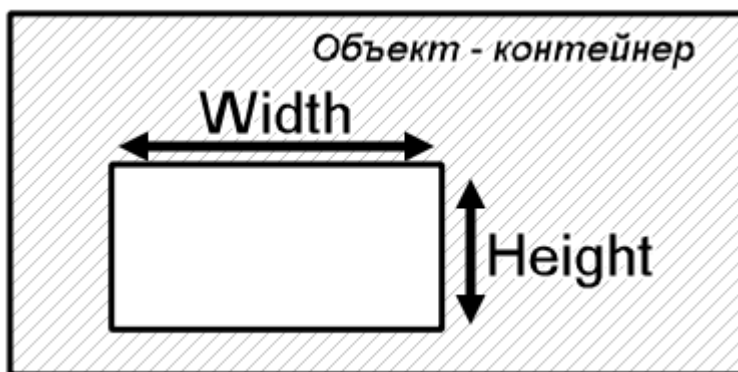
```
Object.Tooltip = "Подача команды MNA_start = true";
```

7.8.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



7.8.1.14. Цвет пера

```
UInt4 PenColor
```

Цвет внешней границы объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.PenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.

```
Object.PenColor = 4278190080;
```

7.8.1.15. Стиль пера

UInt2 PenStyle

Стиль внешней границы объекта.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет пера»	«0»	
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта.

```
Object.PenStyle = 3;
```

7.8.1.16. Толщина пера

```
Float PenWidth
```

Толщина внешней границы объекта. Измеряется в пикселях.

Примеры



ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.PenWidth = 5;
```

7.8.1.17. Файл изображения

```
string ImageFile
```

Путь к файлу изображения относительно папки resources.

7.8.1.18. Активность

```
bool Active
```

Управляет активностью gif-файла.

Значение

true	разрешить воспроизведение анимации
false	запретить воспроизведение анимации

7.8.1.19. Номер кадра

UInt4 CurrentFrameNumber

Позволяет выбрать кадр gif-файла, начиная с которого будет воспроизводиться анимация.

7.8.1.20. Скорость

Float Speed

Управляет скоростью, с которой сменяются кадры анимации.

Значение

0	минимальная скорость смены кадров
от 0 до 1	замедление скорости смены кадров
1	стандартная скорость смены кадров
>1	увеличение скорости воспроизведения кадров

Примеры



ПРИМЕР

Замедлить стандартную скорость смены кадров наполовину.

```
Image.Speed = 0.5;
```

7.8.2. События

7.8.2.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.8.2.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.8.2.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.8.2.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.8.2.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.8.2.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.8.2.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.8.2.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

7.8.2.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события

PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры



ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name", Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o", Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp", Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```



ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

7.8.2.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события

PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры



ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «Ctrl» увеличивать масштаб объекта. В обработчике события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

7.8.2.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки поднятия пальца (пера) по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер поднятия пальца (пера)</p>
PointY	Float	<p>функция, получающая координату точки поднятия пальца (пера) по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер поднятия пальца (пера)</p>

Примеры



ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу OpenSW значение «1». В обработке события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

7.8.2.12. FocusChanged

Изменение фокуса у объекта.

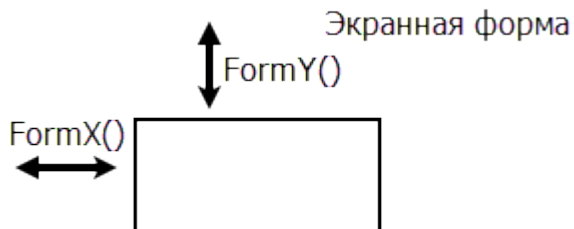
Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

7.8.3. Функции

7.8.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
----------	-----------------------------

Object.Y	значение свойства Y объекта
----------	-----------------------------

Примеры

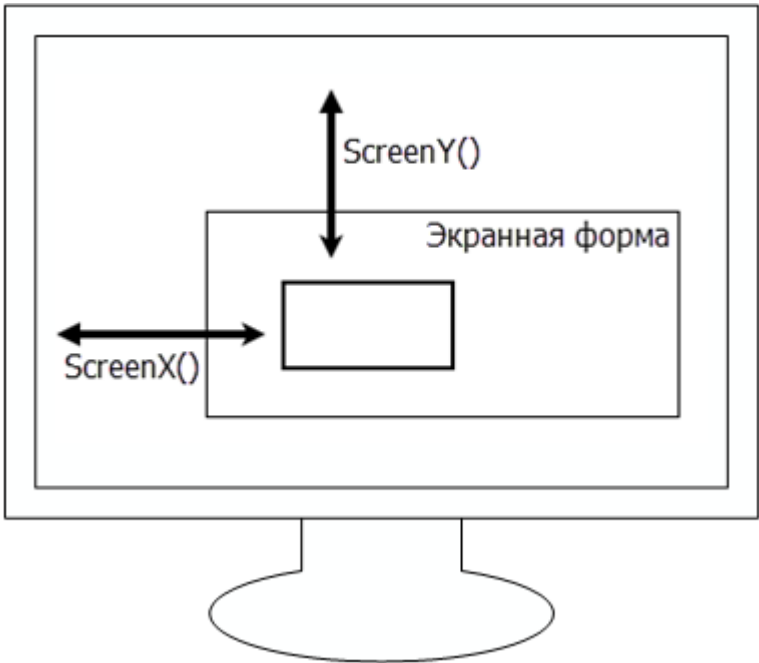
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

7.8.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X,Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

7.9. Точка трансформации

Компонент предназначен для возможности масштабирования и вращения графических объектов относительно данной точки, а не его центра.

Компонент представляет из себя точку, которая размещается на графическом объекте (родительский объект) формы. Масштабирование и вращение родительского объекта будет происходить относительно координат данной точки. Координаты точки вычисляются относительно того объекта, внутри которого точка находится. Значение точки может варьироваться от 0 до 1. Координата точки 0;0 обозначает верхний левый угол объекта.

7.9.1. Свойства

7.9.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

7.9.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

7.9.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

7.9.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

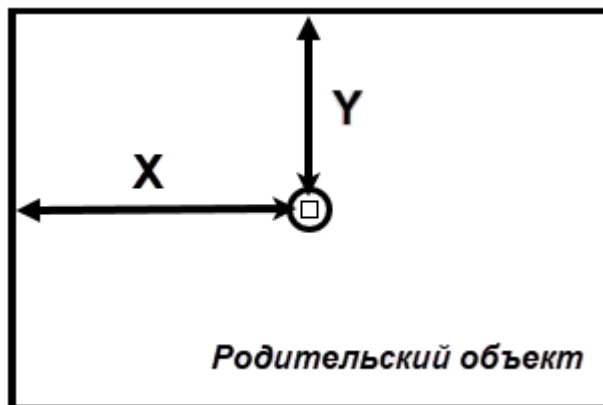
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

7.9.1.5. X и Y

Float X

Float Y

Позиция точки по горизонтали и по вертикали, относительно координат родительского объекта. Могут принимать значения от 0 до 1.



Примеры



ПРИМЕР

Установить позицию точки (0.5; 0.5).

```
OriginPoint.X = 0.5;  
OriginPoint.Y = 0.5;
```

8. Компоненты для построения пользовательского интерфейса

8.1. Индикатор гистограммы

8.1.1. Свойства

8.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.1.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.1.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

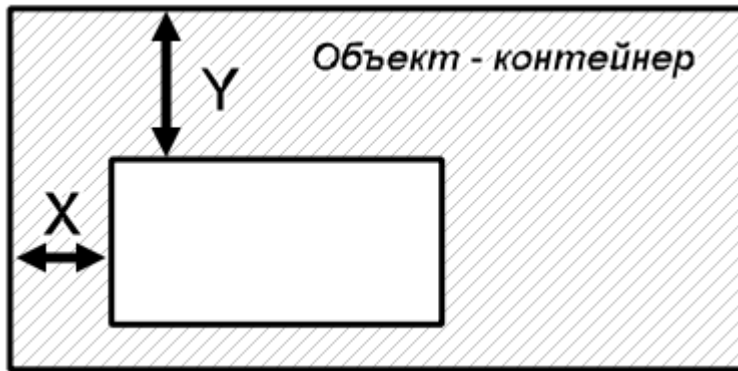
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

8.1.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



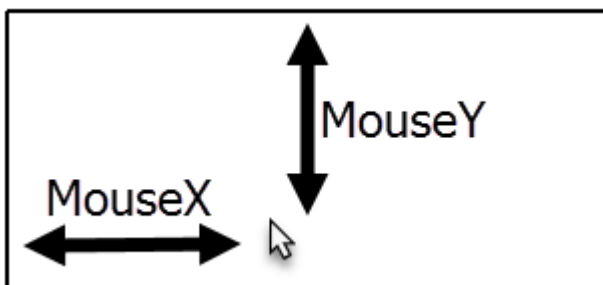
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.1.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.1.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.1.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.1.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.1.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

8.1.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

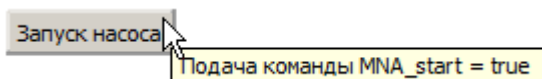
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.1.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

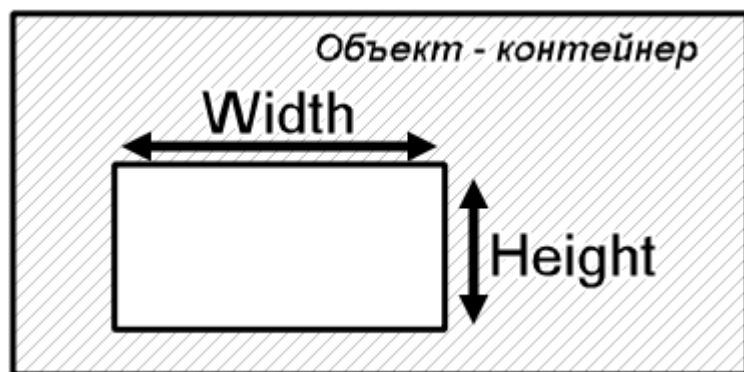
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.1.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



8.1.1.14. Минимальное значение

```
float MinLimit
```

Минимальное значение шкалы гистограммы.

8.1.1.15. Максимальное значение

```
float MaxLimit
```

Максимальное значение шкалы гистограммы.

8.1.1.16. Нижнее пороговое значение

```
float LowValue
```

Значение уровня гистограммы, при понижении которого полоска индикатора изменяет цвет и стиль заливки (Цвет заливки пониженного уровня и Стиль заливки пониженного уровня).

8.1.1.17. Верхнее пороговое значение

```
float HighValue
```

Значение уровня гистограммы, при превышении которого полоска индикатора изменяет цвет и стиль заливки (Цвет заливки превышения уровня и Стиль заливки превышения уровня).

8.1.1.18. Значение

```
float Value
```

Текущее значение уровня на гистограмме.

8.1.1.19. Ориентация

```
Int4 Orientation
```

Ориентация гистограммы.

Значение

0	горизонтальная ориентация гистограммы
1	вертикальная ориентация гистограммы

8.1.1.20. Цвет пера

```
UInt4 PenColor
```

Цвет внешней границы объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.PenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.

```
Object.PenColor = 4278190080;
```

8.1.1.21. Стиль пера

UInt2 PenStyle

Стиль внешней границы объекта.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет пера»	«0»	
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта.

```
Object.PenStyle = 3;
```

8.1.1.22. Толщина пера

Float PenWidth

Толщина внешней границы объекта. Измеряется в пикселях.

Примеры



ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.PenWidth = 5;
```

8.1.1.23. Цвет заливки

UInt4 BrushColor

Цвет внутренней заливки объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Залить объект красным цветом (цвет в шестнадцатеричной системе счисления).

```
Object.BrushColor = 0xffff0000;
```



ПРИМЕР

Залить объект красным цветом (цвет в десятичной системе счисления).




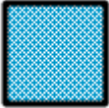



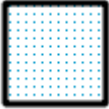

```
Object.BrushColor = 4294901760;
```

8.1.1.24. Стиль заливки

```
UInt2 BrushStyle
```

Заливает объект одним из стилей заливки.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	
«Точечная 1»	«2»	
«Точечная 2»	«3»	
«Точечная 3»	«4»	
«Точечная 4»	«5»	
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



ПРИМЕР

Залить объект сплошным цветом.

```
Object.BrushStyle = 1;
```

8.1.1.25. Цвет заливки полосы

```
UInt4 BarBrushColor
```

Цвет заливки основного уровня гистограммы. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

8.1.1.26. Стиль заливки полосы

```
UInt2 BarBrushStyle
```

Стиль заливки основного уровня гистограммы. Доступные стили аналогичны стилям свойства **Стиль заливки**.

8.1.1.27. Цвет заливки пониженного уровня

```
UInt4 BarLowBrushColor
```

Цвет заливки полосы индикатора, если значение ниже, чем **Нижнее пороговое значение**.

8.1.1.28. Стиль заливки пониженного уровня

```
UInt2 BarLowBrushStyle
```

Стиль заливки полосы индикатора, если значение ниже, чем **Нижнее пороговое значение**. Доступные стили аналогичны стилям свойства **Стиль заливки**.

8.1.1.29. Цвет заливки превышения уровня

```
UInt4 BarHighBrushColor
```

Цвет заливки полосы индикатора, если значение превышает **Верхнее пороговое значение**.

8.1.1.30. Стиль заливки превышения уровня

```
UInt2 BarHighBrushStyle
```

Стиль заливки полосы индикатора, если значение превышает [Верхнее пороговое значение](#). Доступные стили аналогичны стилям свойства [Стиль заливки](#).

8.1.2. События

8.1.2.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```

ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.1.2.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.1.2.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.1.2.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.1.2.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.1.2.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.1.2.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.1.2.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```

ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.1.2.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<div>функция, получающая координату точки нажатия по оси X</div> <div><pre>float8 PointX(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>
PointY	Float	<div>функция, получающая координату точки нажатия по оси Y</div> <div><pre>float8 PointY(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>

Примеры



ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name", Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o", Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp", Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```



ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

8.1.2.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры

ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «Ctrl» увеличивать масштаб объекта. В обработчике события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

8.1.2.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки поднятия пальца (пера) по оси X <div>float8 PointX(uint4)</div> <div>параметр uint4 - порядковый номер поднятия пальца (пера)</div>
PointY	Float	функция, получающая координату точки поднятия пальца (пера) по оси Y <div>float8 PointY(uint4)</div> <div>параметр uint4 - порядковый номер поднятия пальца (пера)</div>

Примеры

ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу OpenSW значение «1». В обработчике события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;
Ellipse_1.Y = 341;
SW_1.OpenSW = 1;
```

8.1.2.12. FocusChanged

Изменение фокуса у объекта.

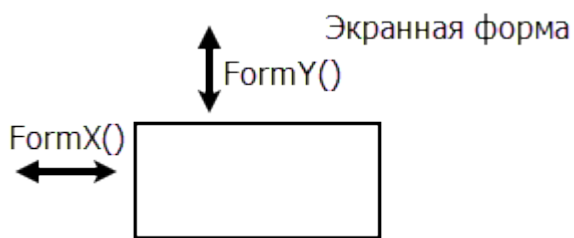
Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.1.3. Функции

8.1.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



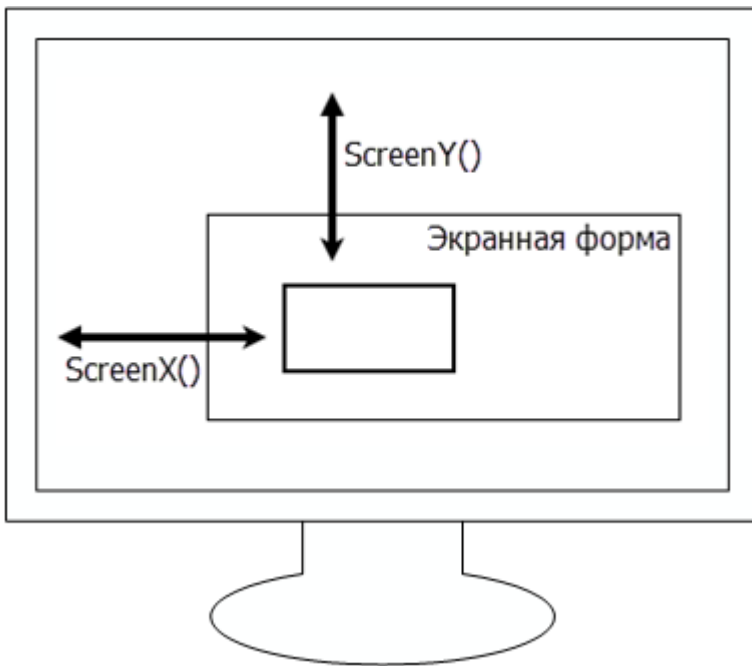
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.1.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.2. Всплывающая подсказка

8.2.1. Свойства

8.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.2.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.2.1.4. Индекс

```
int8 Index
```

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

8.2.1.5. Текст

```
string Text
```

Текстовое содержимое всплывающей подсказки.

Свойство позволяет применять HTML-форматирование для стилизации текста, включая использование жирного начертания, курсива, подчеркивания и изменения цвета.

Примеры



ПРИМЕР

Задать текст подсказки с HTML-форматированием:

```
ToolTip_1.Text = "<b>Статус:</b> <span style='color: green;'>Активен</span><br/>  
<i>Время работы:</i> <u>1234 часов</u><br/>  
<span style='color: red;'><b>Внимание:</b> Требуется обслуживание!</span>";
```

Здесь используются HTML-теги для создания многоуровневого форматирования текста подсказки.



ПРИМЕР

Задать обычный текст для подсказки:

```
ToolTip_1.Text = "Включить/выключить насос";
```

Здесь используется обычный текст без HTML-форматирования для простой и ясной подсказки.

8.2.1.6. Координата X и Координата Y

uint4 X

uint4 Y

Позиция всплывающей подсказки по горизонтали и по вертикали.

Измеряются в пикселях и представляют сдвиг подсказки относительно левого верхнего угла экрана.

Примеры



ПРИМЕР

Задать координату X для всплывающей подсказки:

```
ToolTip_1.X = 150;
```

Задать координату Y для всплывающей подсказки:

```
ToolTip_1.Y = 100;
```

8.2.2. Функции

8.2.2.1. Show

Используется для отображения всплывающей подсказки на экране. Когда этот метод вызывается, подсказка появляется в заданном местоположении, определенном свойствами координат **X** и **Y**. Если координаты **X** и **Y** не заданы, всплывающая подсказка отображается в левом верхнем углу экрана, по умолчанию имея координаты (0; 0).

Синтаксис

```
void Show()
```

Примеры



ПРИМЕР

Отобразить всплывающую подсказку:

```
ToolTip_1.Show();
```

8.2.2.2. Hide

Используется для скрытия всплывающей подсказки. Этот метод убирает подсказку с экрана, делая ее невидимой для пользователя.

Синтаксис

```
void Hide()
```

Примеры



ПРИМЕР

Скрыть всплывающую подсказку:

```
ToolTip_1.Hide();
```

8.3. Текст

Компонент для создания областей с текстом.

8.3.1. Свойства

8.3.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.3.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.3.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.3.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

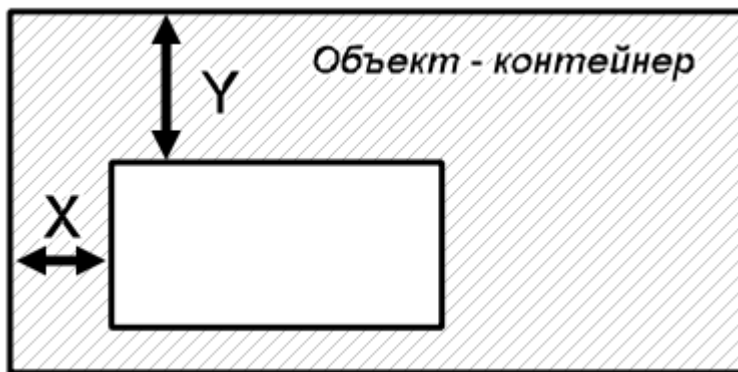
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

8.3.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



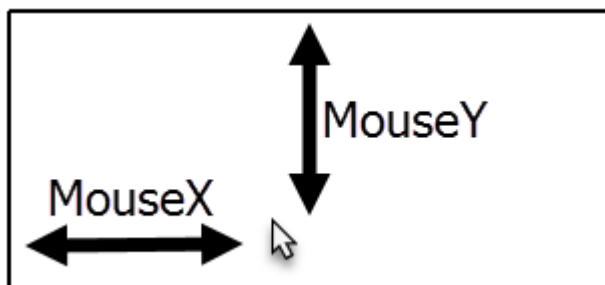
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.3.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.3.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.3.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.3.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.3.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

8.3.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

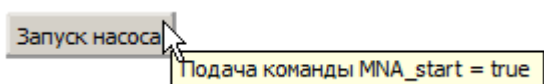
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.3.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

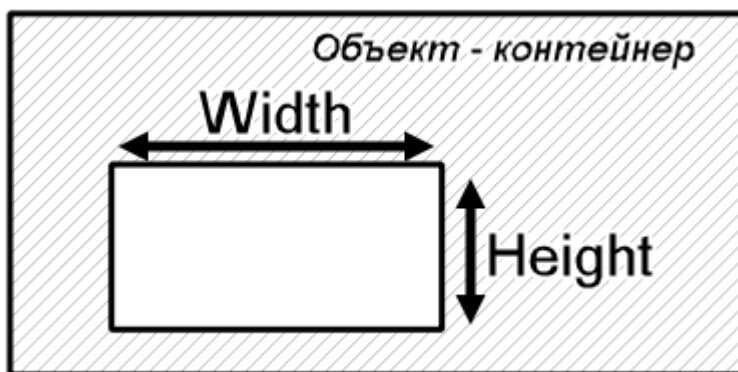
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.3.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



8.3.1.14. Текст

```
string Text
```

Содержимое текстовой области объекта.

8.3.1.15. Шрифт

```
string Font
```

Стиль шрифта. Указывается через системное окно **Выбор шрифта**. После выбора автоматически преобразуется в строку нужного формата.

8.3.1.16. Цвет шрифта

UInt4 FontColor

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.FontColor = 0xffff0000;
```



ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.FontColor = 4294901760;
```

8.3.1.17. Выравнивание текста

Int4 TextAlignment

Значение

33	слева сверху
65	слева снизу
129	слева по центру
34	справа сверху
66	справа снизу
130	справа по центру
36	по центру сверху
68	по центру снизу
132	по центру объекта
40	по всей ширине сверху

72	по всей ширине снизу
136	по всей ширине по центру

8.3.2. События

8.3.2.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.3.2.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.3.2.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.3.2.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.3.2.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.3.2.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.3.2.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```

ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.3.2.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.3.2.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры

ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name",Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o",Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp",Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```

ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

8.3.2.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки нажатия по оси X <div><pre>float8 PointX(uint4)</pre>параметр uint4 - порядковый номер нажатия</div>
PointY	Float	функция, получающая координату точки нажатия по оси Y <div><pre>float8 PointY(uint4)</pre>параметр uint4 - порядковый номер нажатия</div>

Примеры



ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «**Ctrl**» увеличивать масштаб объекта. В обработке события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

8.3.2.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов (« Ctrl », « Shift », « Alt »), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки поднятия пальца (пера) по оси X <pre>float8 PointX(uint4)</pre> параметр uint4 - порядковый номер поднятия пальца (пера)
PointY	Float	функция, получающая координату точки поднятия пальца (пера) по оси Y <pre>float8 PointY(uint4)</pre> параметр uint4 - порядковый номер поднятия пальца (пера)

Примеры



ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу *OpenSW* значение «1». В обработке события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;
Ellipse_1.Y = 341;
SW_1.OpenSW = 1;
```

8.3.2.12. FocusChanged

Изменение фокуса у объекта.

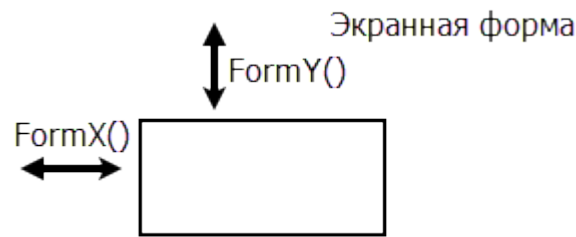
Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.3.3. Функции

8.3.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры

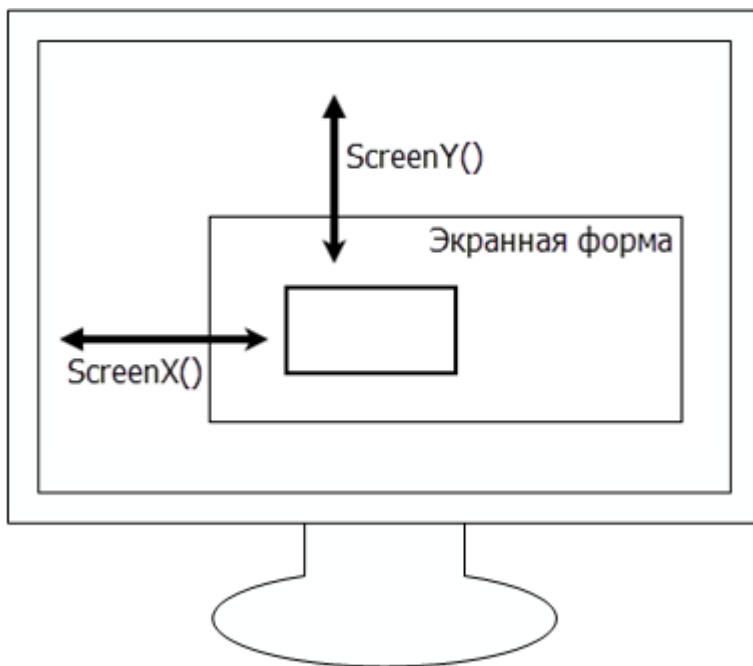
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.3.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.4. Кнопка

8.4.1. Свойства

8.4.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.4.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.4.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.4.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

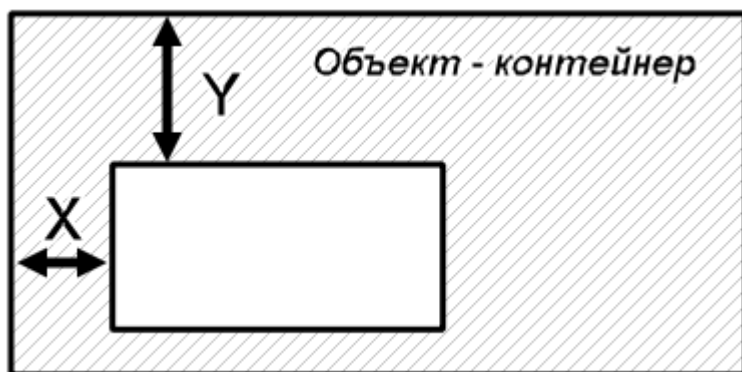
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

8.4.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



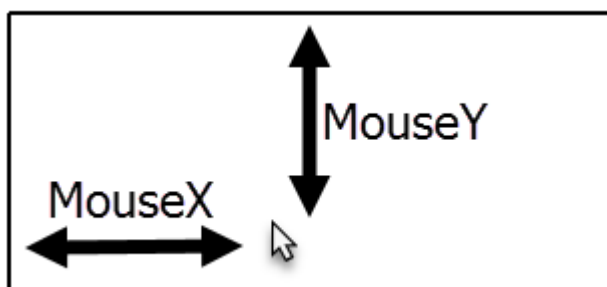
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.4.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.4.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.4.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.4.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.4.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

8.4.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

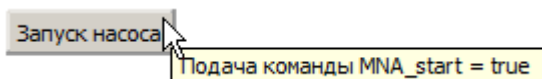
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.4.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры

ПРИМЕР

Задать всплывающую подсказку для объекта.

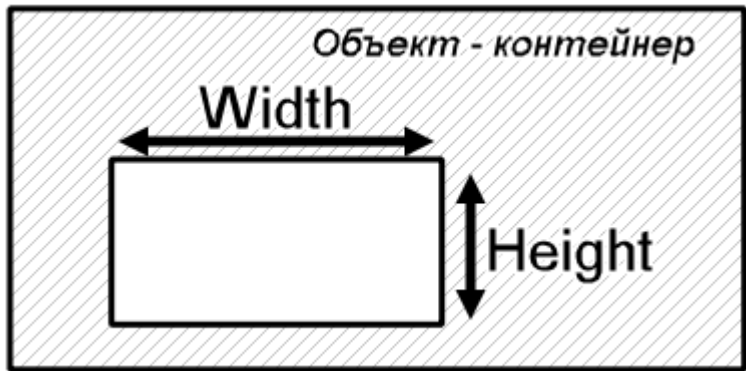
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.4.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



8.4.1.14. Удерживаемая

```
bool Checkable
```

Оставляет кнопку удерживаемой/отпускаемой после нажатия.

Значение

true	удерживать кнопку после нажатия
false	отпускать кнопку после нажатия

8.4.1.15. Текст

```
string Text
```

Содержимое текстовой области объекта.

8.4.1.16. Выравнивание текста

Int4 TextAlignment

Значение

33	слева сверху
65	слева снизу
129	слева по центру
34	справа сверху
66	справа снизу
130	справа по центру
36	по центру сверху
68	по центру снизу
132	по центру объекта
40	по всей ширине сверху
72	по всей ширине снизу
136	по всей ширине по центру

8.4.1.17. Шрифт

string Font

Стиль шрифта. Указывается через системное окно **Выбор шрифта**. После выбора автоматически преобразуется в строку.

8.4.1.18. Цвет шрифта

UInt4 FontColor

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.FontColor = 0xffff0000;
```



ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.FontColor = 4294901760;
```

8.4.1.19. Цвет шрифта при нажатии

UInt4 OnClickFontColor

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.OnClickFontColor = 0xffff0000;
```



ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.OnClickFontColor = 4294901760;
```

8.4.1.20. Цвет шрифта при наведении мыши

UInt4 OnHoverFontColor

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.OnHoverFontColor = 0xffff0000;
```



ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.OnHoverFontColor = 4294901760;
```

8.4.1.21. Цвет заливки

UInt4 BrushColor

Цвет внутренней заливки объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Залить объект красным цветом (цвет в шестнадцатеричной системе счисления).

```
Object.BrushColor = 0xffff0000;
```



ПРИМЕР

Залить объект красным цветом (цвет в десятичной системе счисления).




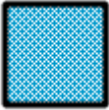



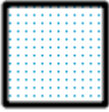

```
Object.BrushColor = 4294901760;
```

8.4.1.22. Стиль заливки


UInt2 BrushStyle

Заливает объект одним из стилей заливки.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	
«Точечная 1»	«2»	
«Точечная 2»	«3»	
«Точечная 3»	«4»	
«Точечная 4»	«5»	
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



ПРИМЕР

Залить объект сплошным цветом.

```
Object.BrushStyle = 1;
```

8.4.1.23. Цвет заливки при нажатии

UInt4 OnClickBrushColor

Цвет внутренней заливки объекта при щелчке левой кнопкой по объекту. Указывается через системное окно **Выбор** цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Залить объект красным цветом (цвет в шестнадцатеричной системе счисления).

```
Object.OnClickBrushColor = 0xffff0000;
```



ПРИМЕР


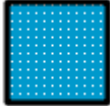
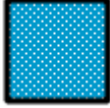
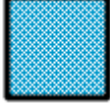
Залить объект красным цветом (цвет в десятичной системе счисления).


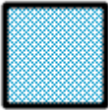
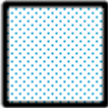
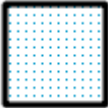

```
Object.OnClickBrushColor = 4294901760;
```

UInt2 OnClickBrushStyle


Заливает объект одним из стилей заливки при щелчке левой кнопкой по объекту.

Значение

Значение	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	
«Точечная 1»	«2»	
«Точечная 2»	«3»	
«Точечная 3»	«4»	

Значение	Значение в коде	Вид границы
«Точечная 4»	«5»	
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



ПРИМЕР

Залить объект сплошным цветом.


```
Object.OnClickBrushStyle = 1;
```



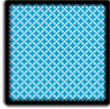

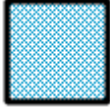
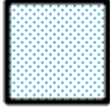


8.4.1.24. Стиль заливки при нажатии

```
UInt2 OnClickBrushStyle
```

Заливает объект одним из стилей заливки при щелчке левой кнопкой по объекту.

Значение

Значение	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	

Значение	Значение в коде	Вид границы
«Точечная 1»	«2»	
«Точечная 2»	«3»	
«Точечная 3»	«4»	
«Точечная 4»	«5»	
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



ПРИМЕР

Залить объект сплошным цветом.

```
Object.OnClickBrushStyle = 1;
```

8.4.1.25. Цвет заливки при наведении мыши

```
UInt4 OnHoverBrushColor
```

Цвет внутренней заливки объекта при вводе курсора в область объекта. Указывается через системное окно **Выбор** цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры

PRIMER

Залить объект красным цветом (цвет в шестнадцатеричной системе счисления).

```
Object.OnHoverBrushColor = 0xffff0000;
```

PRIMER

Залить объект красным цветом (цвет в десятичной системе счисления).


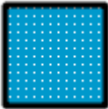
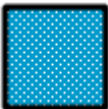
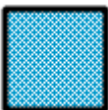

```
Object.OnHoverBrushColor = 4294901760;
```





8.4.1.26. Стиль заливки при наведении мыши

UInt2 OnHoverBrushStyle

Заливает объект одним из стилей заливки при вводе курсора в область объекта.

Значение

Значение	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	
«Точечная 1»	«2»	
«Точечная 2»	«3»	
«Точечная 3»	«4»	
«Точечная 4»	«5»	

Значение	Значение в коде	Вид границы
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



ПРИМЕР

Залить объект сплошным цветом.

```
Object.OnHoverBrushStyle = 1;
```

8.4.1.27. Цвет пера

UInt4 PenColor

Цвет внешней границы объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.PenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.






```
Object.PenColor = 4278190080;
```

8.4.1.28. Стиль пера


UInt2 PenStyle

Стиль внешней границы объекта.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет пера»	«0»	
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры

 ПРИМЕР

Применить стиль границы (короткий пункир) для объекта.

```
Object.PenStyle = 3;
```

8.4.1.29. Толщина пера

Float PenWidth

Толщина внешней границы объекта. Измеряется в пикселях.

Примеры



ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.PenWidth = 5;
```

8.4.1.30. Цвет пера при нажатии

UInt4 OnClickPenColor

Цвет внешней границы объекта при щелчке левой кнопкой по объекту. Указывается через системное окно **Выбор** цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.OnClickPenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.

```
Object.OnClickPenColor = 4278190080;
```

8.4.1.31. Стиль пера при нажатии

UInt2 OnClickPenStyle

Стиль внешней границы объекта при щелчке левой кнопкой по объекту.

Значение

Значение	Значение в коде	Вид границы
«Нет пера»	«0»	

Значение	Значение в коде	Вид границы
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта.

```
Object.OnClickPenStyle = 3;
```

8.4.1.32. Толщина пера при нажатии

Float OnClickPenWidth

Толщина внешней границы объекта при щелчке левой кнопкой по объекту. Измеряется в пикселях.

Примеры



ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.OnClickPenWidth = 5;
```

8.4.1.33. Цвет пера при наведении мыши

UInt4 OnHoverPenColor

Цвет внешней границы объекта при вводе курсора в область объекта. Указывается через системное окно **Выбор** цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления

Примеры



ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.OnHoverPenColor = 0xff000000;
```



ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.




```
Object.OnHoverPenColor = 4278190080;
```



8.4.1.34. Стиль пера при наведении мыши

UInt2 OnHoverPenStyle


Стиль внешней границы объекта при вводе курсора в область объекта.

Значение

Значение	Значение в коде	Вид границы
«Нет пера»	«0»	
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	

Значение	Значение в коде	Вид границы
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта.

```
Object.OnHoverPenStyle = 3;
```

8.4.2. События


8.4.2.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```




ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.4.2.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.4.2.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.4.2.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.4.2.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.4.2.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.4.2.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.4.2.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие

Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры

ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```

ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```

ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.4.2.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<div>функция, получающая координату точки нажатия по оси X</div> <div><pre>float8 PointX(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>
PointY	Float	<div>функция, получающая координату точки нажатия по оси Y</div> <div><pre>float8 PointY(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>

Примеры



ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name", Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o", Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp", Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```



ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

8.4.2.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры

ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «Ctrl» увеличивать масштаб объекта. В обработчике события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

8.4.2.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки поднятия пальца (пера) по оси X <div><pre>float8 PointX(uint4)</pre>параметр uint4 - порядковый номер поднятия пальца (пера)</div>
PointY	Float	функция, получающая координату точки поднятия пальца (пера) по оси Y <div><pre>float8 PointY(uint4)</pre>параметр uint4 - порядковый номер поднятия пальца (пера)</div>

Примеры

ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу OpenSW значение «1». В обработчике события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;
Ellipse_1.Y = 341;
SW_1.OpenSW = 1;
```

8.4.2.12. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.4.2.13. ButtonPressed

Нажатие на кнопку.

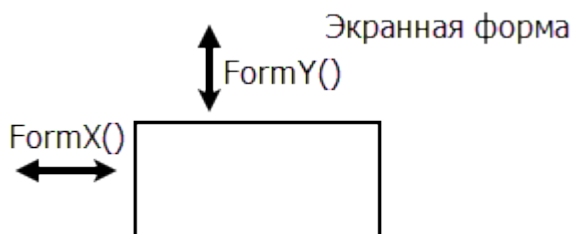
Параметры

checked bool Указывает была ли нажата (true) или отпущена (false) кнопка.

8.4.3. Функции

8.4.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры

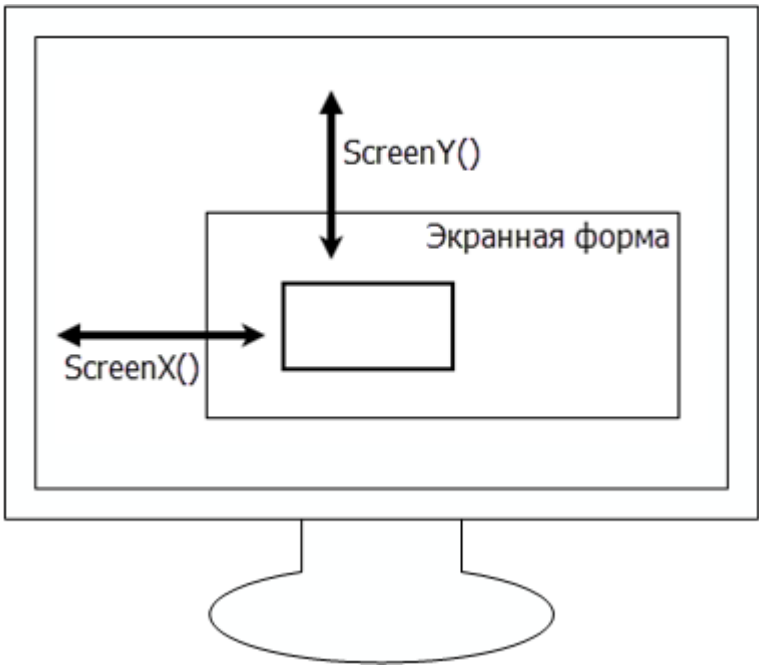
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.4.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.5. Курсор

Компонент **Курсор** предоставляет возможность настройки внешнего вида курсора.

8.5.1. Свойства

8.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.5.1.3. Форма

```
Shape uint1
```

Внешний вид курсора. Позволяет выбирать стандартные формы курсора, предоставляемые операционной системой, или использовать пользовательское изображение для создания уникального курсора (опция «Изображение из файла»).

8.5.1.4. Путь к файлу изображения

```
ImagePath string
```

Путь к файлу изображения, который будет использоваться в качестве курсора. Поддерживаются распространенные форматы изображений, такие как *.png, *.jpg и другие.

Рекомендуется использовать изображения с разрешением «32x32» пикселей, при этом также поддерживаются размеры «16x16», «48x48» и «64x64» пикселей.

Это свойство активно только, когда свойство **Форма** установлено в значение «Изображение из файла».

8.5.1.5. Координата X активного пикселя

HotSpotX int4

Горизонтальное смещение активной точки курсора (точки, которая взаимодействует с элементами на экране) относительно верхнего левого угла изображения курсора. Измеряется в пикселях.

Примеры



ПРИМЕР

Установить горизонтальное смещение активной точки курсора на 10 пикселей.

```
Cursor_1.HotSpotY = 10;
```

Этот код устанавливает горизонтальное смещение активной точки курсора «Cursor_1» на 10 пикселей вправо от верхнего левого угла его изображения.

8.5.1.6. Координата Y активного пикселя

HotSpotY int4

Вертикальное смещение активной точки курсора (точки, которая взаимодействует с элементами на экране) относительно верхнего левого угла изображения курсора. Измеряется в пикселях.

Примеры



ПРИМЕР

Установить вертикальное смещение активной точки курсора на 10 пикселей.

```
Cursor_1.HotSpotY = 10;
```

Этот код смещает активную точку курсора «Cursor_1» на 10 пикселей вниз от верхнего левого угла его изображения

8.6. Системная кнопка

8.6.1. Свойства

8.6.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.6.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.6.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.6.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

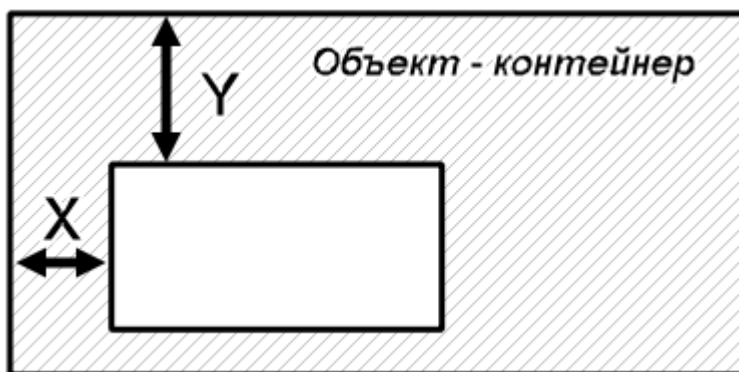
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

8.6.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



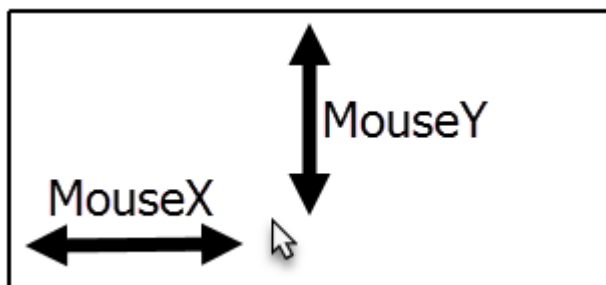
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.6.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.6.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.6.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```




ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.6.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.6.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

8.6.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

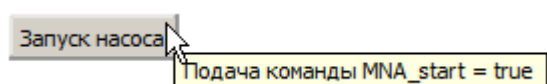
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.6.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

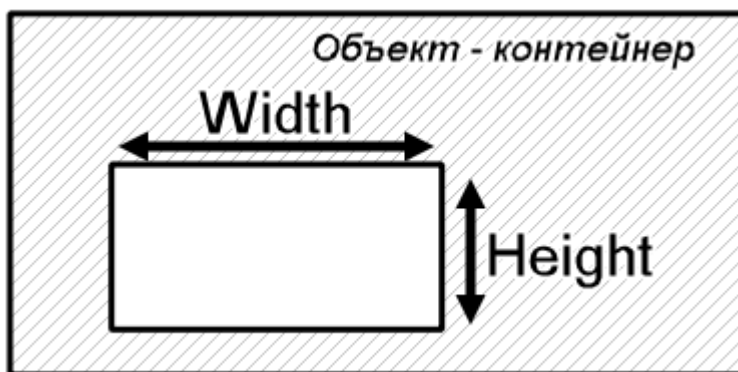
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.6.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



8.6.1.14. Удерживаемая

```
bool Checkable
```

Оставляет кнопку удерживаемой/отпускаемой после нажатия.

Значение

true	удерживать кнопку после нажатия
false	отпускать кнопку после нажатия

8.6.1.15. Текст

```
string Text
```

Содержимое текстовой области объекта.

8.6.1.16. Шрифт

```
string Font
```

Стиль шрифта. Указывается через системное окно **Выбор шрифта**. После выбора автоматически преобразуется в строку.

8.6.1.17. Цвет шрифта

```
UInt4 FontColor
```

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.FontColor = 0xffff0000;
```



ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.FontColor = 4294901760;
```

8.6.2. События

8.6.2.1. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.6.2.2. ButtonPressed

Нажатие на кнопку.

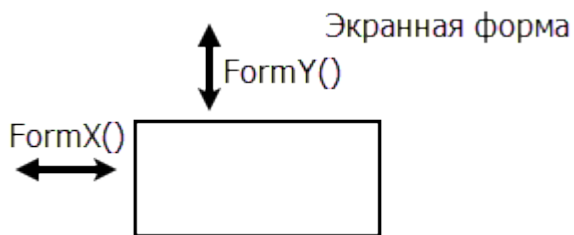
Параметры

checked bool Указывает была ли нажата (true) или отпущена (false) кнопка.

8.6.3. Функции

8.6.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



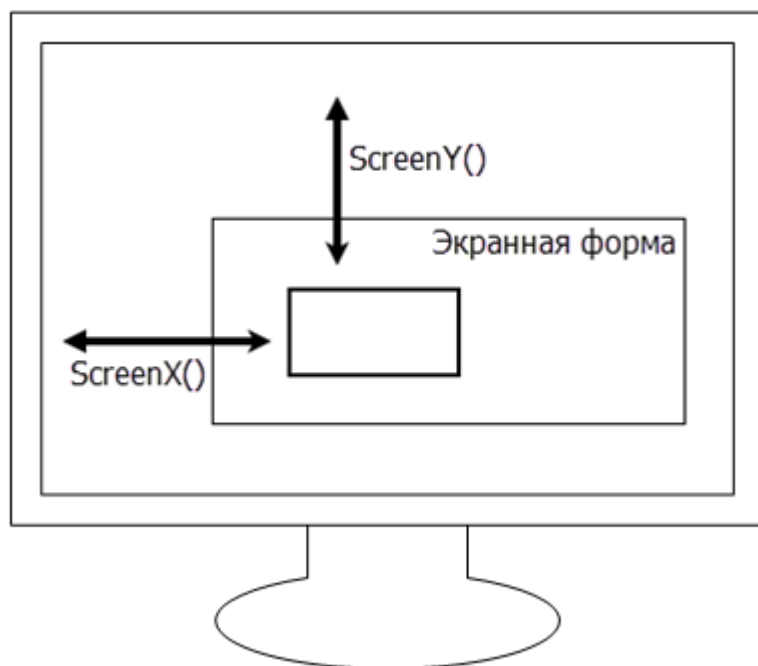
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.6.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.7. Флажок

8.7.1. Свойства

8.7.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.7.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.7.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.7.1.4. Индекс

```
int8 Index
```

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

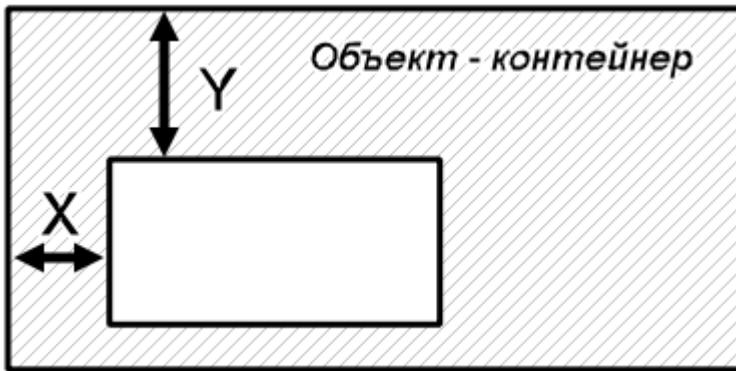
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

8.7.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



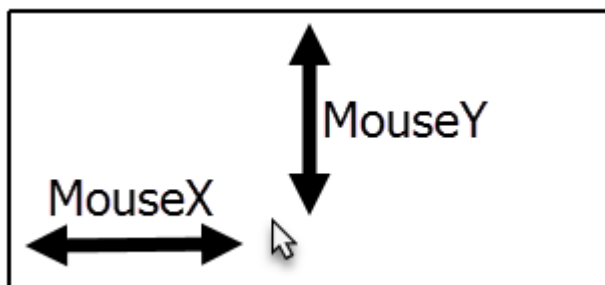
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.7.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.7.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.7.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.7.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.7.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

8.7.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

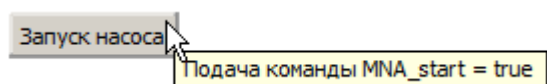
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.7.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

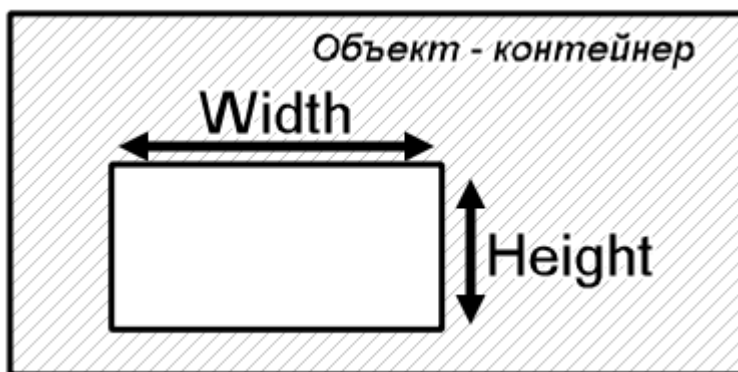
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.7.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



8.7.1.14. Текст

```
string Text
```

Содержимое текстовой области объекта.

8.7.1.15. Шрифт

```
string Font
```

Стиль шрифта. Указывается через системное окно **Выбор шрифта**. После выбора автоматически преобразуется в строку.

8.7.1.16. Цвет шрифта

```
UInt4 FontColor
```

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры

ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.FontColor = 0xffff0000;
```

ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.FontColor = 4294901760;
```

8.7.1.17. Состояние

```
bool State
```

Состояние флажка.

Значение

true	флажок включен
false	флажок выключен

8.7.2. События

8.7.2.1. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.7.2.2. StateChanged

Изменение состояния флажка.

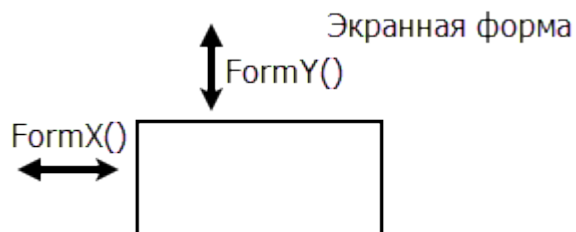
Параметры

state bool указывает был ли установлен флаг (true) или снят (false).

8.7.3. Функции

8.7.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



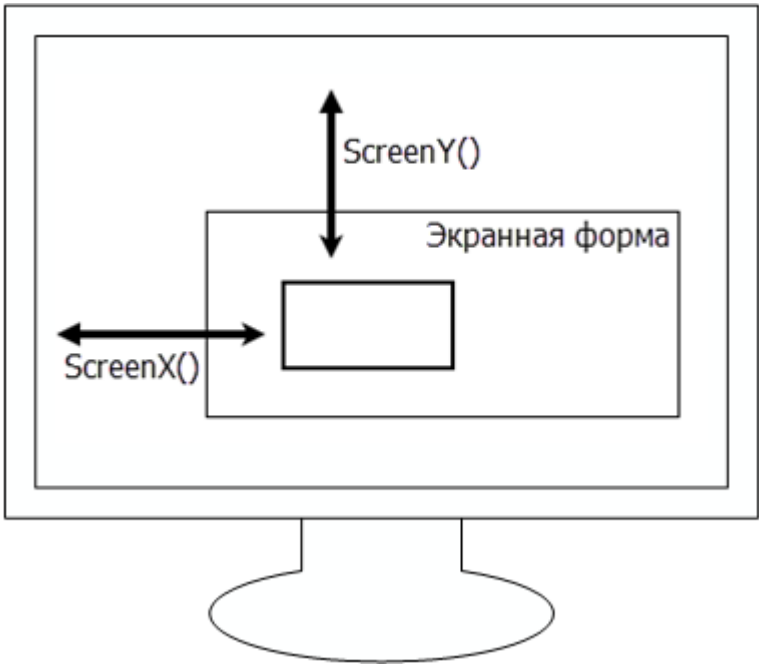
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.7.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X,Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры

ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```


8.8. Поле ввода

8.8.1. Свойства

8.8.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.8.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.8.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.8.1.4. Индекс

```
int8 Index
```

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

8.8.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



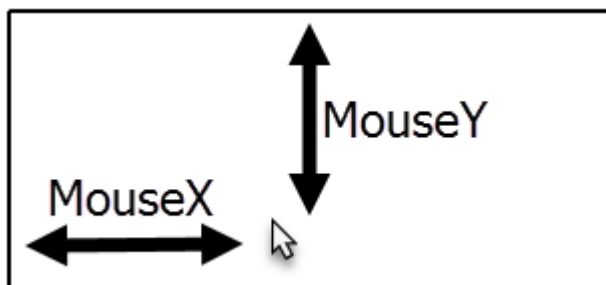
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.8.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.8.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.8.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.8.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.8.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

8.8.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

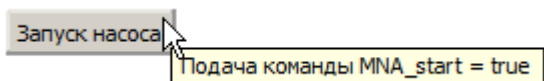
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.8.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

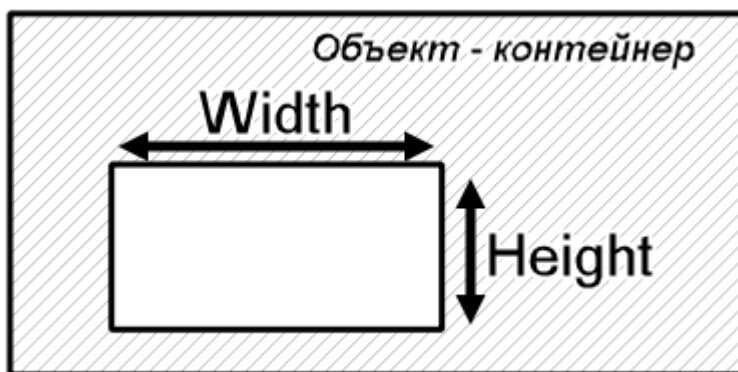
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.8.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



8.8.1.14. Текст

```
string Text
```

Содержимое текстовой области объекта.

8.8.1.15. Шрифт

```
string Font
```

Стиль шрифта. Указывается через системное окно **Выбор шрифта**. После выбора автоматически преобразуется в строку нужного формата.

8.8.1.16. Цвет шрифта

UInt4 FontColor

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.FontColor = 0xffff0000;
```



ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.FontColor = 4294901760;
```

8.8.1.17. Выравнивание текста

Int4 TextAlignment

Значение

33	слева сверху
65	слева снизу
129	слева по центру
34	справа сверху
66	справа снизу
130	справа по центру
36	по центру сверху
68	по центру снизу
132	по центру объекта
40	по всей ширине сверху

72	по всей ширине снизу
136	по всей ширине по центру

8.8.1.18. Скрывать ввод

```
bool HideInput
```

Режим маскировки вводимого текста.

Значение

true	вводимые символы маскируются
false	вводимые символы не маскируются

8.8.1.19. Текст заполнителя

```
string PlaceholderText
```

Текст, который отображается в пустом текстовом поле. Чтобы отображался текст заполнителя, свойство Текст должно быть незаполненным.

Поле ввода

8.8.2. События

8.8.2.1. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.8.2.2. EditFinished

Завершение редактирования текста снятием фокуса с поля ввода или нажатием «Enter».

Параметры

text string новый текст в поле ввода

8.8.2.3. TextChanged

Редактирование текста.



Параметры

text string новый текст в поле ввода

8.8.2.4. KeyPress

Нажатие клавиши или сочетания клавиш в поле ввода. Чтобы обратиться к параметрам события, используйте ключевое слово event.

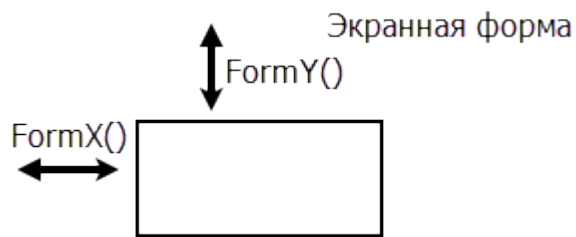
Параметры

KeyCode	UInt4	код нажатой клавиши
		<div>  ПРИМЕР При нажатии клавиши «Enter» записать текст из поля ввода в текстовое поле. </div> <pre>if (event.KeyCode == 82) {Text_1.Text = TextEdit_1.Text;}</pre>
Modifiers	Int4	код нажатой клавиши-модификатора
Repeated	bool	флаг удержания клавиши или сочетания любых клавиш. Не становится «true» при удержании клавиш-модификаторов: «Ctrl», «Alt», «Shift»
		<div>  ПРИМЕР При удержании клавиши «F5» записать текст из поля ввода в текстовое поле. </div> <pre>if (event.KeyCode == 55 && event.Repeated == true) {Text_1.Text = TextEdit_1.Text;}</pre>
Text	string	текст нажатой клавиши

8.8.3. Функции

8.8.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



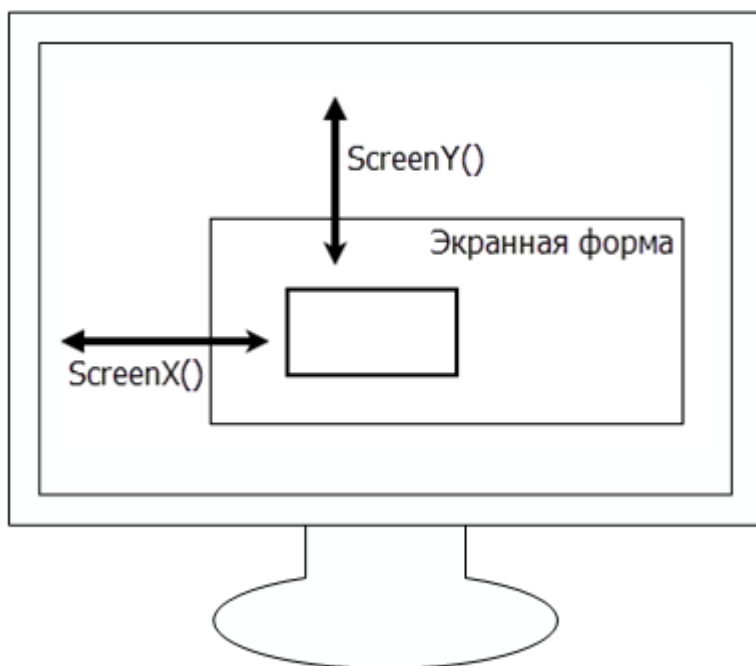
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.8.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.9. Индикатор прогресса

8.9.1. Свойства

8.9.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.9.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.9.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.9.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

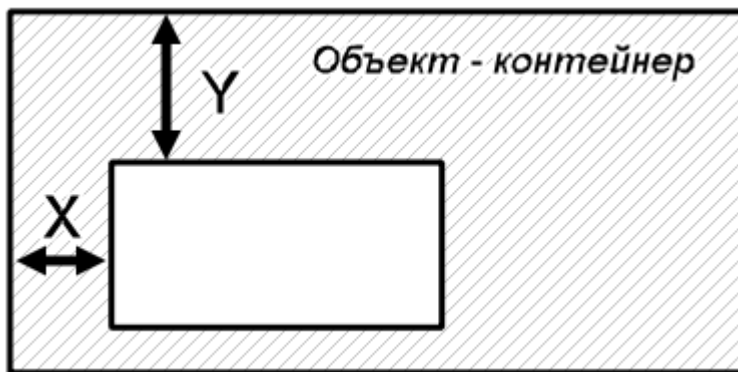
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

8.9.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



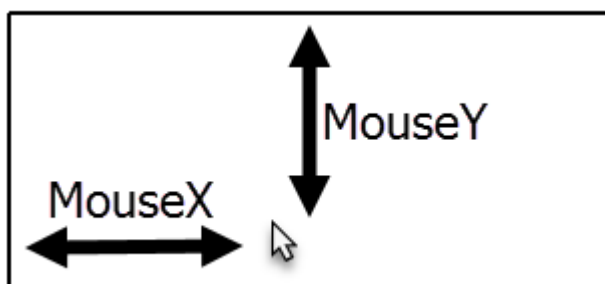
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.9.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.9.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.9.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```




ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.9.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.9.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

8.9.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

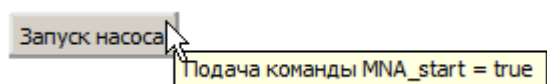
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.9.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

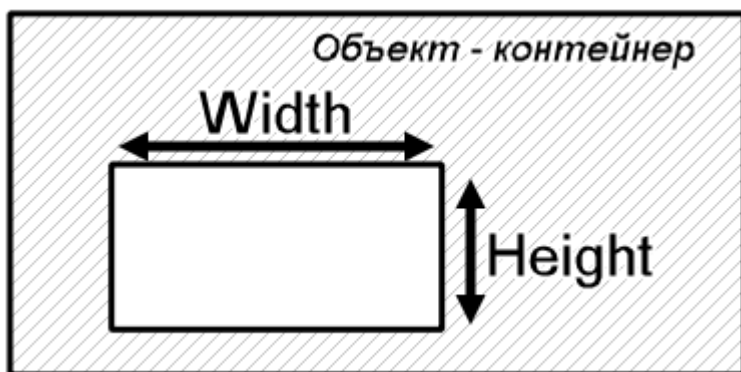
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.9.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

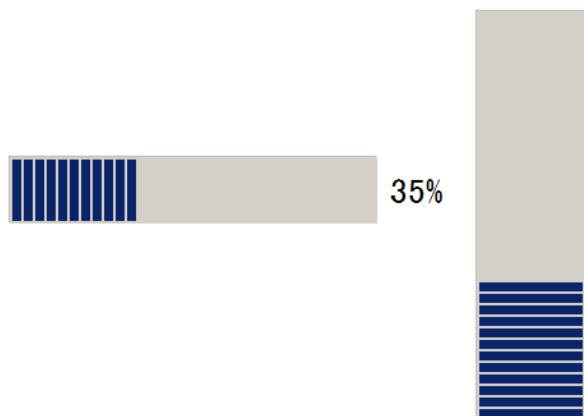
Ширина и высота объекта в пикселях.



8.9.1.14. Ориентация

```
Int4 Orientation
```

Расположение индикатора прогресса.



Значение

0	индикатор прогресса ориентирован горизонтально и у него есть цифровое поле
2	индикатор прогресса ориентирован вертикально, без цифрового поля

8.9.1.15. Минимальное значение

Int4 MinLimit

Значение, которое будет соответствовать 0% заполненности шкалы прогресса.

8.9.1.16. Максимальное значение

Int4 MaxLimit

Значение, которое будет соответствовать 100% заполненности шкалы прогресса.

8.9.1.17. Значение

Int4 Value

Текущее значение индикатора прогресса.

8.9.1.18. Показывать текст

bool ShowText

Позволяет отобразить/скрыть текущее значение прогресса справа от индикатора.

Значение

true	текущее значение прогресса отображается
false	текущее значение прогресса не отображается

8.9.1.19. Шрифт

string Font

Стиль шрифта. Указывается через системное окно **Выбор шрифта**. После выбора автоматически преобразуется в строку нужного формата.

8.9.1.20. Цвет шрифта

UInt4 FontColor

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.FontColor = 0xffff0000;
```



ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.FontColor = 4294901760;
```

8.9.2. События

8.9.2.1. FocusChanged

Изменение фокуса у объекта.

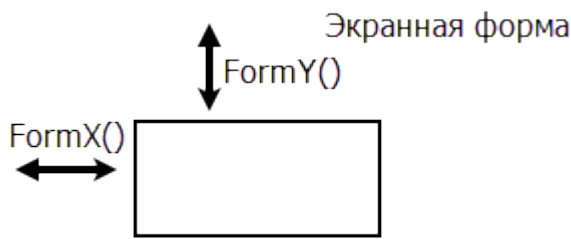
Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.9.3. Функции

8.9.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры

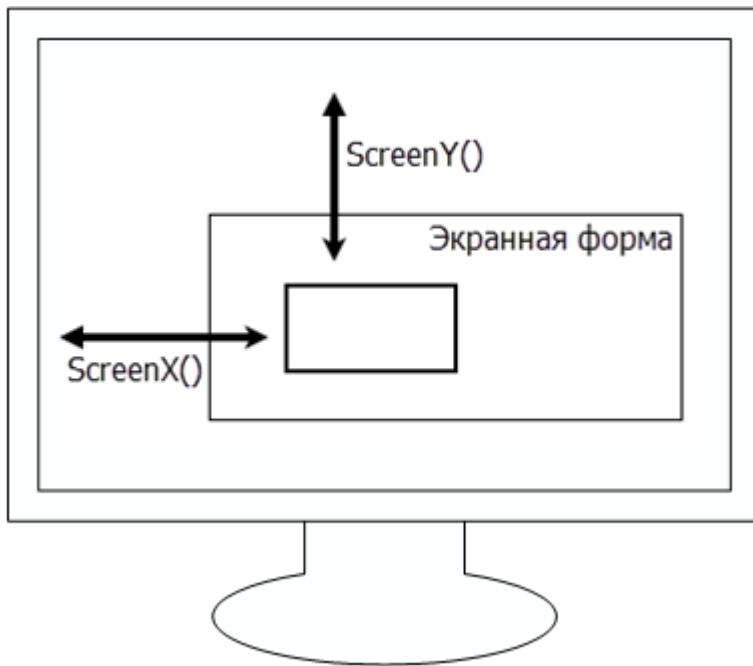
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.9.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.10. Ползунок

8.10.1. Свойства

8.10.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.10.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.10.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.10.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

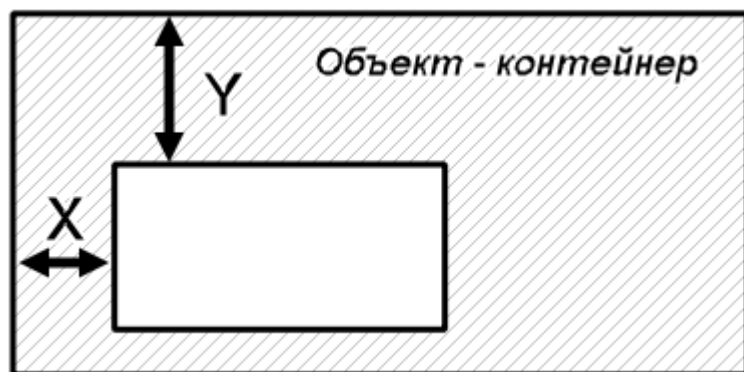
```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

8.10.1.5. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



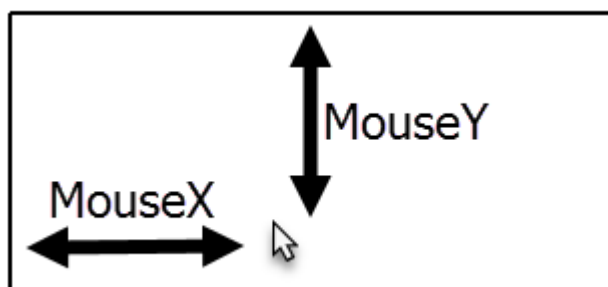
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.10.1.6. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.10.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.10.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.10.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.10.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

8.10.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

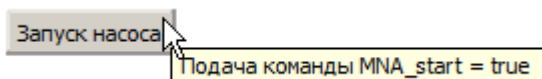
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.10.1.12. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

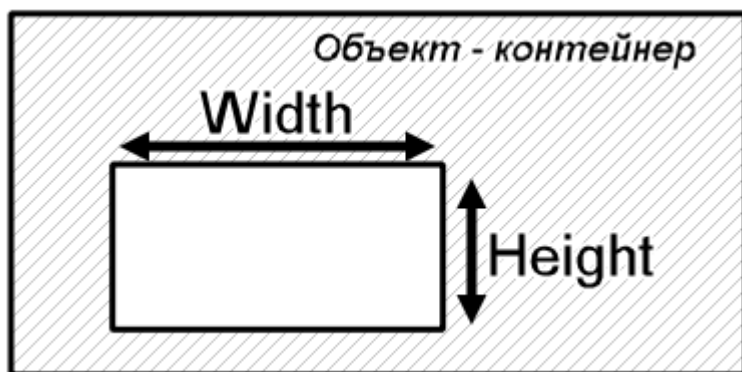
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.10.1.13. Ширина и Высота

```
float Width
```

```
float Height
```

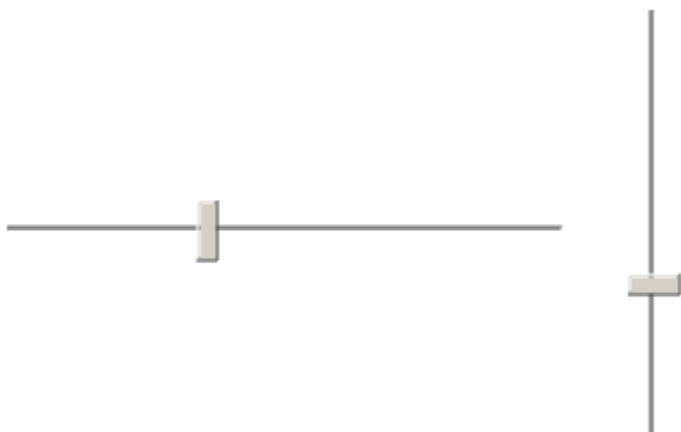
Ширина и высота объекта в пикселях.



8.10.1.14. Ориентация

```
Int4 Orientation
```

Расположение ползунка.



Значение

1	ползунок ориентирован горизонтально
2	ползунок ориентирован вертикально

8.10.1.15. Минимальное значение

Int4 MinLimit

Значение, которое соответствует минимальному положению ползунка.

8.10.1.16. Максимальное значение

Int4 MaxLimit

Значение, которое соответствует максимальному положению ползунка.

8.10.1.17. Значение

Int4 Value

Текущее значение ползунка.

8.10.2. События

8.10.2.1. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.10.2.2. ValueChanged

Событие при изменении положения ползунка.

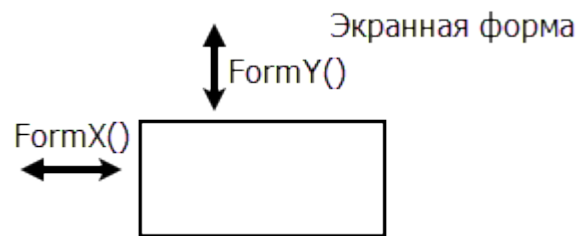
Параметры

value	Int4	новое значение положения ползунка
-------	------	-----------------------------------

8.10.3. Функции

8.10.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X,Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры

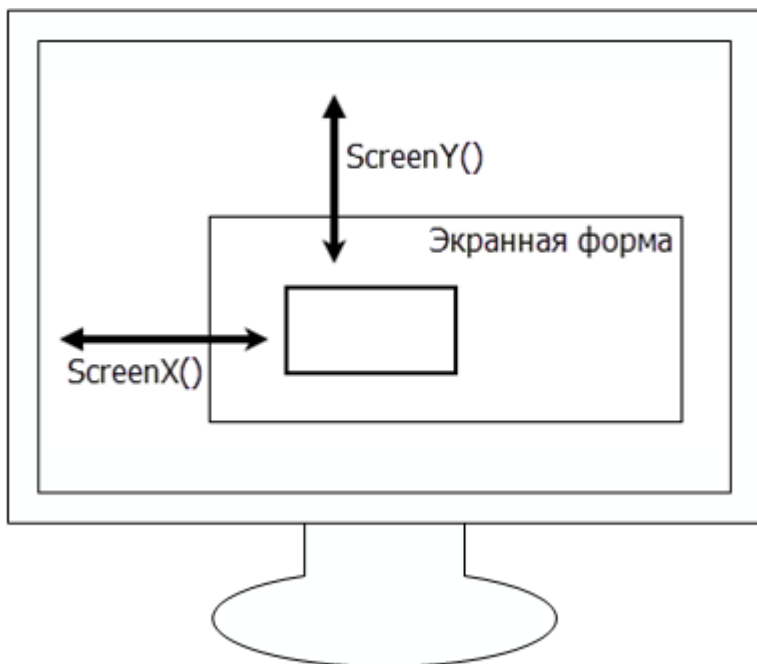
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.10.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.11. Выпадающий список

8.11.1. Свойства

8.11.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.11.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

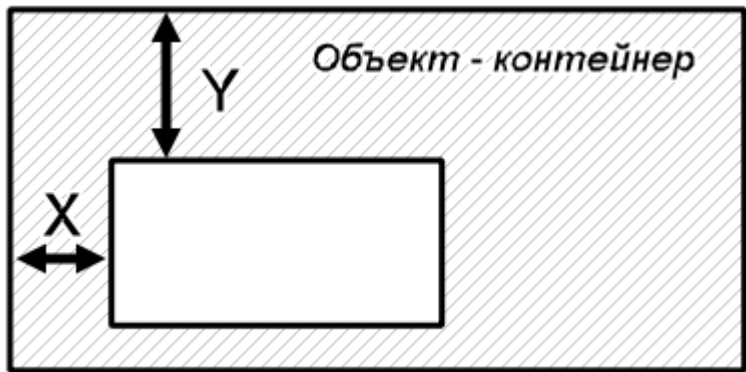
«1»	Одиночный объект.
>«1»	Массив соответствующей размерности.

8.11.1.3. Координата X и Координата Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

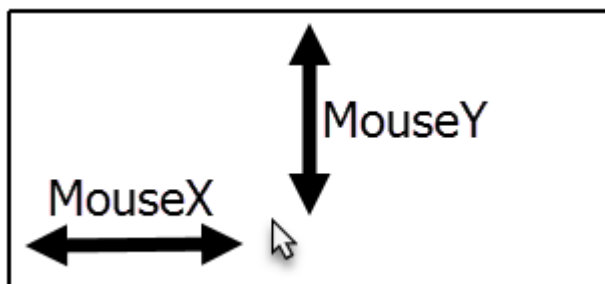
8.11.1.4. Z-значение

Float ZValue

Порядок отображения объекта на экране. Чем выше значение свойства **Z-значение**, тем объект отображается выше, перекрывая объекты с меньшими значениями свойства **Z-значение**.

8.11.1.5. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.11.1.6. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.11.1.7. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```

**ПРИМЕР**

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```

**ПРИМЕР**

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.11.1.8. Отражение

```
uint1 Flip
```

Отражение объекта относительно горизонтальной и вертикальной осей.

Значение

«Без отражения »	Объект отображается без изменений.
«По горизонтали»	Объект отражается по горизонтали, создавая зеркальное отображение относительно вертикальной оси.
«По вертикали»	Объект отражается по вертикали, создавая зеркальное отображение относительно горизонтальной оси
«По горизонтали и вертикали»	Объект отражается одновременно и по горизонтали, и по вертикали, создавая зеркальное отображение относительно обеих осей.

8.11.1.9. Видимость

```
bool Visible
```

Видимость объекта на форме.

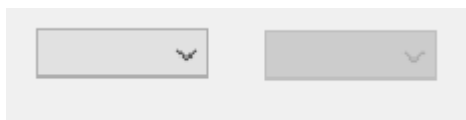
Значение

true	Объект виден.
false	Объект скрыт.

8.11.1.10. Включено

bool Enabled

Активность объекта.



Значение

true	Объект активен.
false	Объект заблокирован.

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

8.11.1.11. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	Фокус на объекте.
false	Фокус вне объекта.

Примеры



ПРИМЕР

Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.11.1.12. Редактируемый

```
bool Editable
```

Доступность поля выпадающего списка для редактирования.

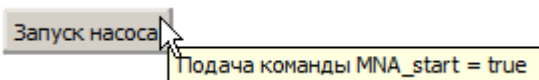
Значение

true	Позволяет пользователю вводить текст в поле выпадающего списка.
false	Запрещает пользователю вводить текст в поле выпадающего списка. В этом случае пользователь может только выбирать значения из предложенного списка.

8.11.1.13. Всплывающая подсказка

```
string Tooltip
```

Подсказка, всплывающая при наведении курсора на объект.



ПРИМЕР

Задать всплывающую подсказку для объекта.

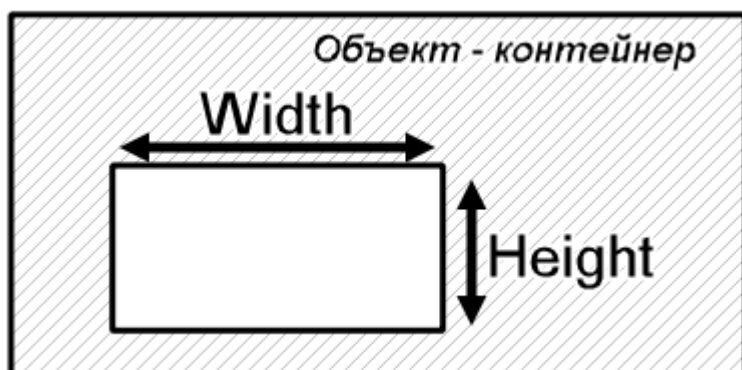
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.11.1.14. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



8.11.1.15. Текст

```
string Text
```

Текст, который виден до нажатия на выпадающий список и появления списка доступных элементов для выбора.

8.11.1.16. Выбранный индекс

```
uint4 SelectedIndex
```

Индекс элемента из выпадающего списка, который выбран в данный момент в компоненте **Выпадающий список**.

Если ни один элемент из списка не выбран, значением будет «-1».

8.11.1.17. Количество видимых элементов

```
uint4 MaxVisibleItems
```

Максимальное количество элементов, отображаемых одновременно в выпадающем списке.

8.11.1.18. Режим вставки

```
uint1 InsertPolicy
```

Режим вставки новых элементов в выпадающем списке в режиме редактирования.

Свойство **Редактируемый** должно быть установлено в значение «true», чтобы разрешить редактирование выпадающего списка пользователем.

Значение

«Без вставки»	Новые элементы не добавляются в список.
«В начало»	Новые элементы вставляются в начало списка.
«Замена выбранного элемента»	Новый элемент заменяет выбранный элемент.
«В конец»	Новые элементы добавляются в конец списка.
«После выбранного элемента»	Новый элемент добавляется после выбранного элемента.
«Перед выбранным элементом»	Новый элемент добавляется перед выбранным элементом.
«По алфавиту»	Новый элемент добавляется в алфавитном порядке.

8.11.1.19. Шрифт

```
string Font
```

Стиль шрифта. Указывается через системное окно **Выбор шрифта**. После выбора автоматически преобразуется в строку нужного формата.

8.11.1.20. Цвет шрифта

```
UInt4 FontColor
```

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.FontColor = 0xffff0000;
```



ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.FontColor = 4294901760;
```

8.11.2. События

8.11.2.1. Activated

Активация (выбор) элемента пользователем в выпадающем списке.

Параметры

index	int4	Индекс выбранного элемента, который был активирован (выбран) пользователем.
-------	------	-----------------------------------------------------------------------------

8.11.2.2. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	Флаг наличия фокуса на объекте.
----------	------	---------------------------------

8.11.2.3. SelectedChanged

Изменение выбранного элемента в выпадающем списке.

Параметры

selected	int4	Индекс элемента, который стал выбранным.
----------	------	------------------------------------------

8.11.2.4. TextChanged

Изменение текста в поле выпадающего списка.

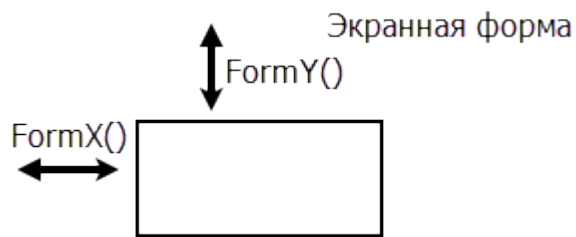
Параметры

text	string	Текст, который был изменен в поле выпадающего списка.
------	--------	-------------------------------------------------------

8.11.3. Функции

8.11.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



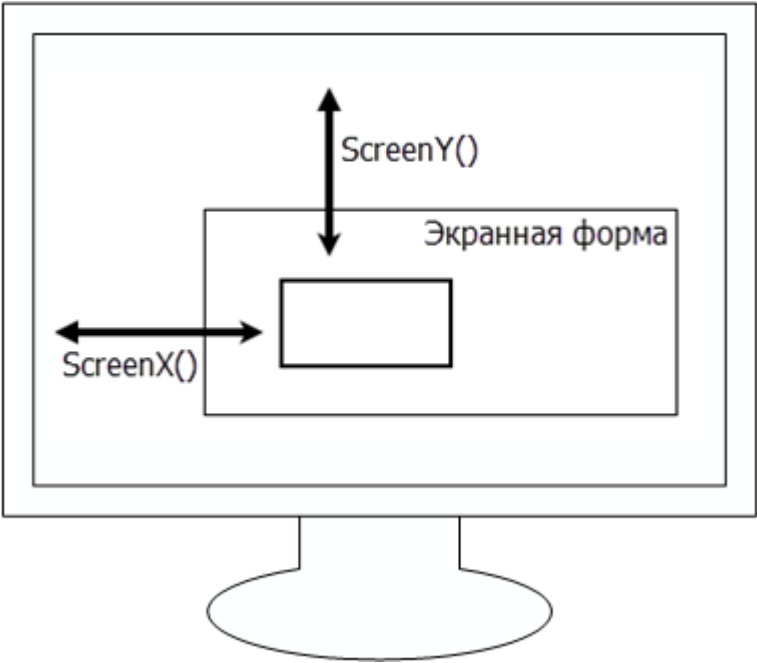
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.11.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X,Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры

ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.11.3.3. Clear

Используется для полной очистки элементов из выпадающего списка.

Синтаксис

```
void Clear()
```

Примеры



ПРИМЕР

Удалить все элементы из выпадающего списка.

```
ComboBox_1.Clear();
```

8.11.3.4. GetItem

Используется для извлечения текста элемента из выпадающего списка по его индексу.

Синтаксис

```
string GetItem(int4 itemIndex)
```

Параметры

itemIndex	Индекс элемента в списке, для которого нужно получить текст.
-----------	--------------------------------------------------------------

Возвращаемое значение

Возвращает текст элемента, соответствующего указанному индексу в списке.

Примеры



ПРИМЕР

Поместить в текстовое поле значение элемента выпадающего списка по его индексу:

```
selectedText: string = ComboBox_1.GetItem(2);  
Text_1.Text= selectedText;
```

Этот код извлекает текст элемента из выпадающего списка «ComboBox_1» по индексу «2» и устанавливает его в текстовое поле «Text_1».

8.11.3.5. IndexOf

Используется для получения индекса элемента в выпадающем списке по его тексту.

Синтаксис

```
int4 IndexOf(string itemText)
```


Параметры

itemText	Текст элемента, для которого необходимо найти индекс в списке.
----------	----------------------------------------------------------------

Возвращаемое значение

Возвращает индекс элемента, соответствующего указанному тексту в выпадающем списке. Если элемент не найден, возвращается значение «-1».

Примеры

 **ПРИМЕР**

Получить индекс элемента в выпадающем списке по его тексту:

```
index: int4 = ComboBox_1.IndexOf("Оператор");
```

Этот код находит индекс элемента в выпадающем списке «**ComboBox_1**» с текстом «**Оператор**» и сохраняет его в переменной «**index**».

8.11.3.6. SetItem

Используется для установки текста определенного элемента в выпадающем списке по его индексу.

Синтаксис

```
void SetItem(int4 itemIndex, string itemText)
```

Параметры

itemIndex	Индекс устанавливаемого элемента.
itemText	Текст, который будет установлен для выбранного элемента.

Примеры



ПРИМЕР

Заменить текст для элемента из выпадающего списка с индексом «2»:

```
ComboBox_1.SetItem(2, "Новый текст");
```

Этот код изменяет текст элемента с индексом «2» в выпадающем списке «ComboBox_1» на «Новый текст».

8.11.3.7. AddItem

Используется для добавления нового элемента в выпадающий список.

Синтаксис

```
void AddItem(string ItemText)
```

Параметры

itemText	Текст, который будет установлен добавляемому элементу.
----------	--------------------------------------------------------

Примеры



ПРИМЕР

Добавить новый элемент в выпадающий список:

```
ComboBox_1.AddItem("Новый элемент");
```

Этот код добавляет новый элемент с текстом «Новый элемент» в выпадающий список «ComboBox_1».

8.11.3.8. RemoveItem

Используется для удаления элемента из выпадающего списка по его индексу.

Синтаксис

```
void RemoveItem(uint4 itemIndex)
```

Параметры

itemIndex	Индекс элемента, который нужно удалить из списка.
-----------	---------------------------------------------------

Примеры



ПРИМЕР

Удалить элемент с индексом «2» из выпадающего списка:

```
ComboBox_1.RemoveItem(2);
```

Этот код удаляет элемент с индексом «2» из выпадающего списка «ComboBox_1».

8.12. Фрейм

Область, в которой можно динамически открывать формы. Чтобы открыть форму во фрейме следует в обработчике событий выбрать команду **Открыть во фрейме**, указать фрейм, в котором откроется форма, и форму.

8.12.1. Свойства

8.12.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.12.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.12.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.12.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

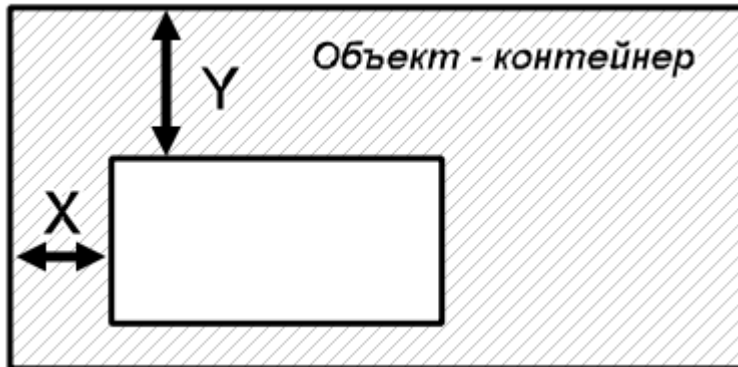
В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



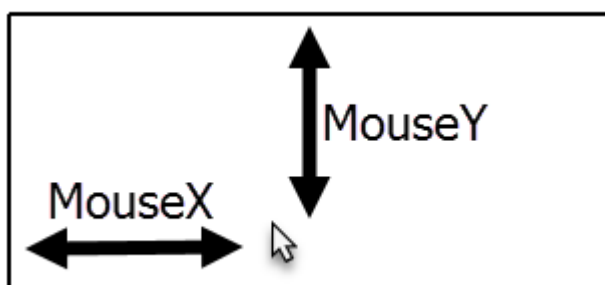
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.12.1.5. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



**ПРИМЕЧАНИЕ**

Доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.12.1.6. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры

**ПРИМЕР**

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```

**ПРИМЕР**

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.12.1.7. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры

ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```

ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```

ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.12.1.8. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.12.1.9. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

8.12.1.10. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

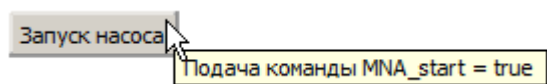
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)  
{  
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);  
}
```

8.12.1.11. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

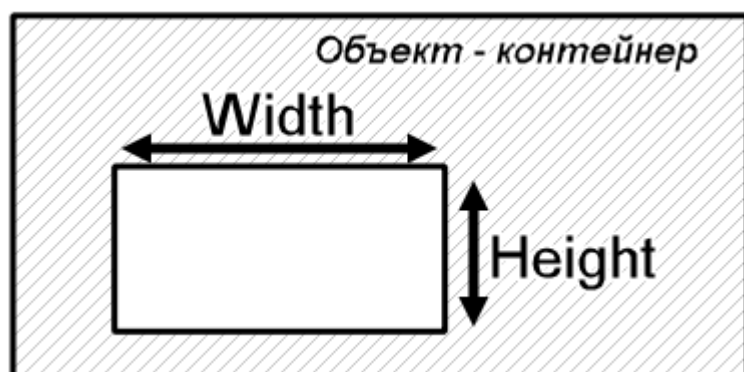
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.12.1.12. Ширина и Высота

float Width

float Height

Ширина и высота объекта в пикселях.



8.12.1.13. Цвет пера

UInt4 PenColor

Цвет внешней границы объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры

ПРИМЕР

Задать черный цвет пера в шестнадцатеричной системе счисления.

```
Object.PenColor = 0xff000000;
```

ПРИМЕР

Задать черный цвет пера в десятичной системе счисления.

```
Object.PenColor = 4278190080;
```

8.12.1.14. Стиль пера

UInt2 PenStyle

Стиль внешней границы объекта.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет пера»	«0»	
«Сплошная линия»	«1»	
«Штриховая линия»	«2»	
«Пунктирная линия»	«3»	
«Штрих-пунктирная линия»	«4»	
«Штрих-пунктирная линия с двумя точками»	«5»	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта.

```
Object.PenStyle = 3;
```

8.12.1.15. Толщина пера

Float PenWidth

Толщина внешней границы объекта. Измеряется в пикселях.

Примеры



ПРИМЕР

Задать толщину пера 5 пикселей.

```
Object.PenWidth = 5;
```

8.12.1.16. Цвет заливки

UInt4 BrushColor

Цвет внутренней заливки объекта. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Залить объект красным цветом (цвет в шестнадцатеричной системе счисления).

```
Object.BrushColor = 0xffff0000;
```



ПРИМЕР

Залить объект красным цветом (цвет в десятичной системе счисления).




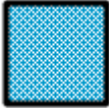

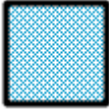



```
Object.BrushColor = 4294901760;
```


8.12.1.17. Стил ь заливки

UInt2 BrushStyle

Заливает объект одним из стилей заливки.

Значение

Значение в дизайнере	Значение в коде	Вид границы
«Нет заливки»	«0»	
«Сплошная заливка»	«1»	
«Точечная 1»	«2»	
«Точечная 2»	«3»	
«Точечная 3»	«4»	
«Точечная 4»	«5»	
«Точечная 5»	«6»	
«Точечная 6»	«7»	
«Точечная 7»	«8»	
Горизонтальные линии	«9»	

Примеры



ПРИМЕР

Залить объект сплошным цветом.

```
Object.BrushStyle = 1;
```

8.12.1.18. Ручное управление масштабом

```
bool OverrideScaling
```

Разрешает/запрещает использование функций [ScaleContentTo](#), [SetContentScale](#) и [MoveViewport](#) фрейма.

Значение

true	разрешено ручное управление масштабом
false	запрещено ручное управление масштабом

Примеры



ПРИМЕР

При установке флажка включить режим ручного масштабирования.

```
if (Checkbox_1.State == true)
{Frame_1.OverrideScaling = true;}
```

8.12.1.19. Отображать полосы прокрутки

```
bool ShowScrollBars
```

Отображает/скрывает полосы прокрутки для формы, открытой во фрейме.

Значение

true	полосы прокрутки отображаются
false	полосы прокрутки скрыты

Примеры



ПРИМЕР

Отображать/скрывать полосы прокрутки при нажатии пробела.

```
if (event.KeyCode == 84)
    Frame_1.ShowScrollBars = !Frame_1.ShowScrollBars;
```

8.12.1.20. Перемещение мышью

```
bool MoveByMouse
```

Позволяет/запрещает перетаскивать форму, открытую во фрейме, мышью.

Значение

true	перемещение формы мышью разрешено
false	перемещение формы мышью запрещено

8.12.2. События

8.12.2.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.12.2.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.12.2.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.12.2.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.12.2.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.12.2.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.12.2.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.12.2.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие

Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

8.12.2.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры

ПРИМЕР

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name",Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o",Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp",Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```

ПРИМЕР

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

8.12.2.10. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<div>функция, получающая координату точки нажатия по оси X</div> <div><pre>float8 PointX(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>
PointY	Float	<div>функция, получающая координату точки нажатия по оси Y</div> <div><pre>float8 PointY(uint4)</pre></div> <div>параметр uint4 - порядковый номер нажатия</div>

Примеры



ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «**Ctrl**» увеличивать масштаб объекта. В обработке события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

8.12.2.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов (« Ctrl », « Shift », « Alt »), которые были зажаты при возникновении события
PointX	Float	функция, получающая координату точки поднятия пальца (пера) по оси X <pre>float8 PointX(uint4)</pre> параметр uint4 - порядковый номер поднятия пальца (пера)
PointY	Float	функция, получающая координату точки поднятия пальца (пера) по оси Y <pre>float8 PointY(uint4)</pre> параметр uint4 - порядковый номер поднятия пальца (пера)

Примеры



ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу *OpenSW* значение «1». В обработке события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;
Ellipse_1.Y = 341;
SW_1.OpenSW = 1;
```

8.12.2.12. FocusChanged

Изменение фокуса у объекта.

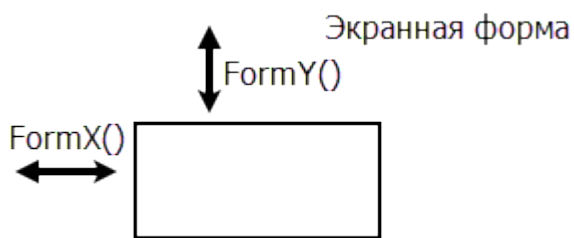
Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.12.3. Функции

8.12.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



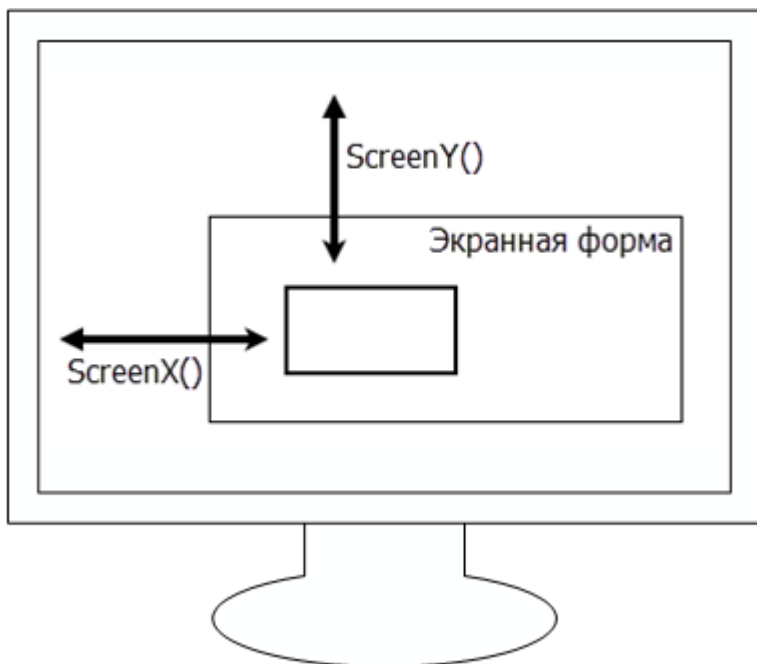
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.12.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.13. Дерево

8.13.1. Свойства

8.13.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

8.13.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

8.13.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

8.13.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

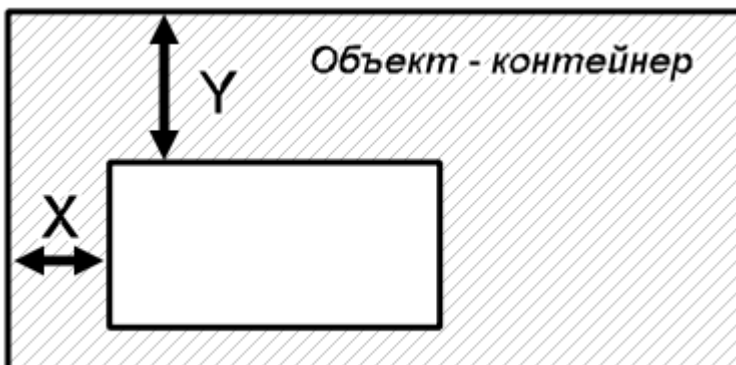
В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



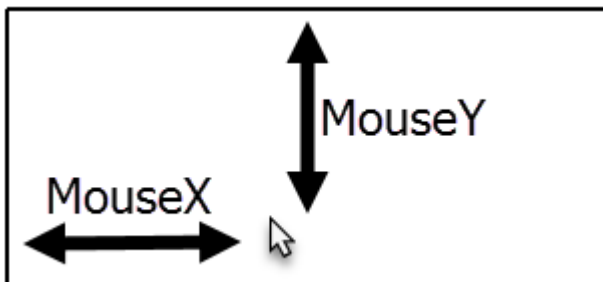
ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

8.13.1.5. Координата курсора X и Координата курсора Y

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта.

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

8.13.1.6. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

8.13.1.7. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

8.13.1.8. Видимость

bool Visible

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

8.13.1.9. Включено

bool Enabled

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

8.13.1.10. Фокус ввода

bool Focus

Переводит фокус на объект/снимает фокус с объекта.

Значение

true	фокус на объекте
false	фокус вне объекта

Примеры



ПРИМЕР

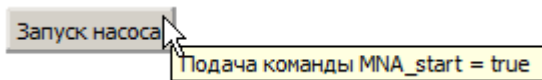
Записать значение из поля ввода в сигнал при снятии фокуса с объекта.

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

8.13.1.11. Всплывающая подсказка

```
string Tooltip
```

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

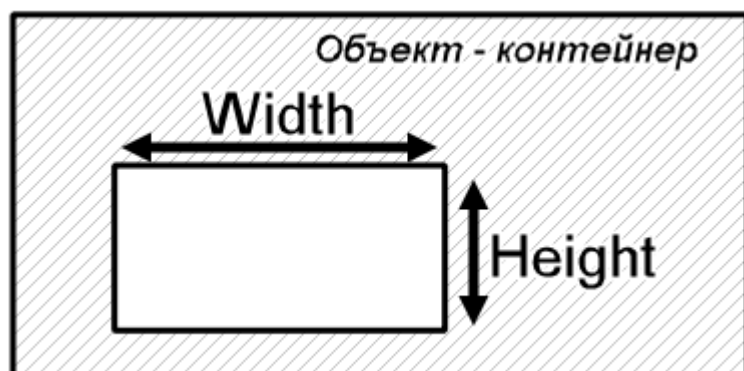
```
Object.Tooltip = "Подача команды MNA_start = true";
```

8.13.1.12. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



8.13.1.13. Шрифт

```
string Font
```

Стиль шрифта. Указывается через системное окно **Выбор шрифта**. После выбора автоматически преобразуется в строку.

8.13.1.14. Цвет шрифта

```
UInt4 FontColor
```

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать красный цвет шрифта в шестнадцатеричной системе счисления.

```
Object.FontColor = 0xffff0000;
```



ПРИМЕР

Задать красный цвет шрифта в десятичной системе счисления.

```
Object.FontColor = 4294901760;
```

8.13.1.15. Цвет фона

```
UInt4 BackgroundColor
```

Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления.

Примеры



ПРИМЕР

Задать серый цвет фона в шестнадцатеричной системе счисления.

```
Object.BackgroundColor = 0x9CA2A3;
```



ПРИМЕР

Задать серый цвет фона в десятичной системе счисления.

```
Object.BackgroundColor = 10265251;
```

8.13.1.16. Видимость рамки

```
bool FrameVisible
```

Видимость рамки, обрамляющей список форм.

Значение

true	рамка видна
false	рамка скрыта

8.13.1.17. Число столбцов

```
UInt4 ColumnCount
```

Отображает/скрывает полосы прокрутки для формы, открытой во фрейме.

8.13.2. События

8.13.2.1. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметры

PointsCount	uint4	количество нажатий
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<div>функция, получающая координату точки нажатия по оси X</div> <div>float8 PointX(uint4)</div> <div>параметр uint4 - порядковый номер нажатия</div>
PointY	Float	<div>функция, получающая координату точки нажатия по оси Y</div> <div>float8 PointY(uint4)</div> <div>параметр uint4 - порядковый номер нажатия</div>

Примеры

PRIMER

При перетягивании объекта *Tank* на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события **TouchMove** объекта *Tank* пропишите передаваемые значения.

```
Tank_1.DragNDrop_1.SetData("name",Str.ToString(rez345.Name.Value));
Tank_1.DragNDrop_1.SetData("h2o",Str.ToString(rez345.H2O.Value));
Tank_1.DragNDrop_1.SetData("temp",Str.ToString(rez345.Temp.Value));
Tank_1.DragNDrop_1.DoDragDrop();
```

PRIMER

Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y. В обработчике события **TouchMove** формы пропишите:

```
TextEdit_1 = Str.ToString(event.PointX);
TextEdit_2 = Str.ToString(event.PointY);
```

8.13.2.2. TouchPress

Нажатие на объект.

Параметры

PointsCount	uint4	количество нажатий
-------------	-------	--------------------

KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки нажатия по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>
PointY	Float	<p>функция, получающая координату точки нажатия по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер нажатия</p>

Примеры



ПРИМЕР

При нажатии на объект *SW* и зажатии клавиши «Ctrl» увеличивать масштаб объекта. В обработке события **TouchPress** объекта *SW* пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

8.13.2.3. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (пера) в любом месте экрана.

Параметры

PointsCount	uint4	количество поднятий пальца (пера)
KeyboardModifiers	Int4	коды клавиш-модификаторов («Ctrl», «Shift», «Alt»), которые были зажаты при возникновении события
PointX	Float	<p>функция, получающая координату точки поднятия пальца (пера) по оси X</p> <pre>float8 PointX(uint4)</pre> <p>параметр uint4 - порядковый номер поднятия пальца (пера)</p>
PointY	Float	<p>функция, получающая координату точки поднятия пальца (пера) по оси Y</p> <pre>float8 PointY(uint4)</pre> <p>параметр uint4 - порядковый номер поднятия пальца (пера)</p>

Примеры



ПРИМЕР

При поднятии пальца (пера) после нажатия на объект *SW* нужно сместить кнопку и присвоить сигналу OpenSW значение «1». В обработчике события **TouchRelease** объекта типа *SW* пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

8.13.2.4. ItemPressed

Нажатие левой кнопки мыши на элемент дерева.

Параметры

itemid	string	идентификатор выбранного элемента дерева
column	Int4	номер столбца выбранного элемента дерева

8.13.2.5. FocusChanged

Изменение фокуса у объекта.

Параметры

hasFocus	bool	флаг наличия фокуса на объекте
----------	------	--------------------------------

8.13.2.6. ItemClicked

Щелчок по элементу дерева.

Параметры

itemid	string	идентификатор выбранного элемента дерева
column	int4	номер столбца выбранного элемента дерева

8.13.2.7. ItemDoubleClicked

Двойной щелчок левой кнопкой мыши по элементу дерева.

Параметры

itemid	string	идентификатор выбранного элемента дерева
column	int4	номер столбца выбранного элемента дерева

8.13.2.8. ItemExpanded

Разворачивание узлов элемента дерева.

Параметры

itemid	string	идентификатор выбранного элемента дерева
--------	--------	------------------------------------------

8.13.2.9. ItemCollapsed

Сворачивание узлов элемента дерева.

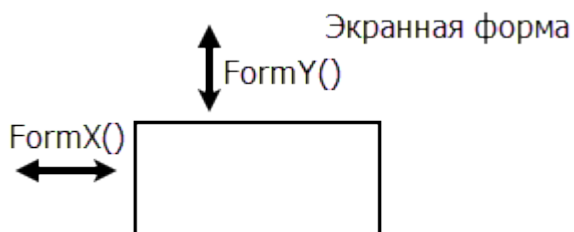
Параметры

itemid	string	идентификатор выбранного элемента дерева
--------	--------	------------------------------------------

8.13.3. Функции

8.13.3.1. FormX и FormY

Вычисляют координаты объекта относительно экранной формы, на которой размещен объект. Координаты измеряются в пикселях.



Синтаксис

```
FormX(Object.X, Object.Y)
```

```
FormY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



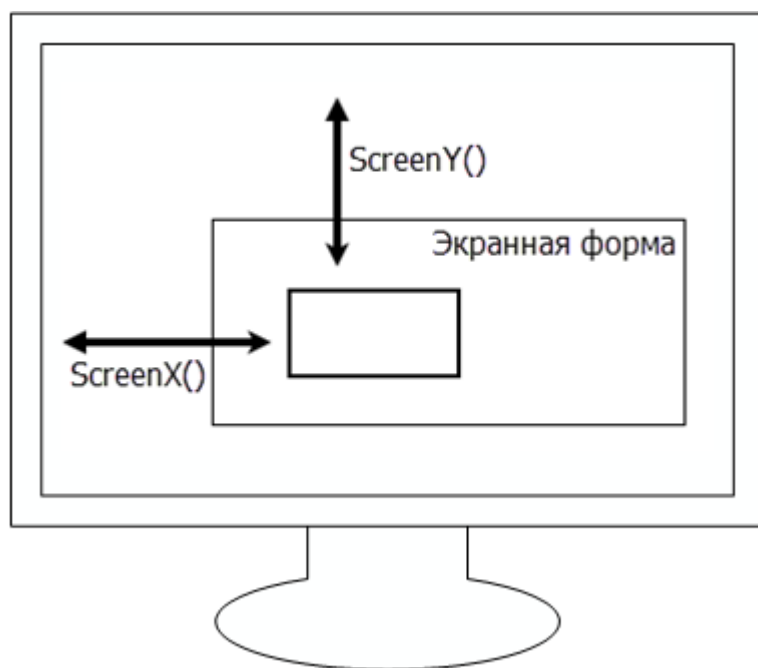
ПРИМЕР

Записать координаты X и Y объекта относительно экранной формы в поля ввода.

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);  
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

8.13.3.2. ScreenX и ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	значение свойства X объекта
Object.Y	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода.

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

8.13.3.3. AddItem

Добавляет новый элемент в список форм.

Синтаксис

```
string AddItem (string ItemID, string ItemText)
```

Параметры

ItemID	string	идентификатор родительского элемента. Для вставки корневого элемента списка оставьте значение пустым. Элемент будет вставлен в конце списка дочерних/корневых элементов
ItemText	string	имя элемента

Примеры



ПРИМЕР

Добавить новый элемент в состав дочерних.

```
item_1: var = Tree_1.AddItem(item_0, "Задвижка 1");
```

8.13.3.4. InsertItem

Добавляет новый дочерний элемент в указанное место.

Синтаксис

```
string InsertItem (string ItemID, string ItemText, int4 n)
```

Параметры

ItemID	string	идентификатор родительского элемента. Для вставки корневого элемента списка оставьте значение пустым. Элемент будет вставлен в конце списка дочерних/корневых элементов
ItemText	string	имя элемента
n	int4	порядковый номер дочернего элемента внутри родительского элемента, после которого будет вставлен новый элемент

Примеры

ПРИМЕР

Добавить новый элемент в состав дочерних.

```
Tree_1.InsertItem(item_0, "Задвижка 1", 3);
```

8.13.3.5. ChildCount

Возвращает число дочерних элементов.

Синтаксис

```
string ChildCount (string ItemID)
```

Параметры

ItemID	string	идентификатор элемента
--------	--------	------------------------

Примеры

ПРИМЕР

Вывести число дочерних элементов в текстовом поле.

```
ch_count = Tree_1.ChildCount(item_1);
TextEdit_2.Text = Str.ToString(ch_count);
```



ПРИМЕР

8.13.3.6. MoveItem

Перемещает элемент в указанное место.

Синтаксис

```
void MoveItem (string ItemID, string ItemID, n)
```

Параметры

ItemID	string	идентификатор родительского элемента
ItemID	string	идентификатор перемещаемого элемента
n	int4	порядковый номер дочернего элемента внутри родительского элемента, после которого будет вставлен перемещаемый элемент. Если указанное число превышает количество дочерних элементов, перемещаемый элемент вставится в конец списка, если меньше 0 - в начало списка.

Примеры



ПРИМЕР

Переместить элемент на второе место в списке.

```
Tree_1.MoveItem(item1, item5, 1);
```

8.13.3.7. RemoveItem

Удаляет элемент.

Синтаксис

```
void RemoveItem (string ItemID)
```

Параметры

ItemID	string	идентификатор элемента
--------	--------	------------------------

Примеры



ПРИМЕР

Удалить выделенный элемент.

```
Tree_1.RemoveItem(SelectedItem);
```

8.13.3.8. RemoveChildren

Удаляет дочерние элементы.

Синтаксис

```
void RemoveChildren (string ItemID)
```

Параметры

ItemID	string	идентификатор родительского элемента
--------	--------	--------------------------------------

Примеры



ПРИМЕР

Удалить дочерние элементы выделенного элемента.

```
Tree_1.RemoveChildren(SelectedItem);
```

8.13.3.9. SetColumnName

Устанавливает имя столбца.

Синтаксис

```
void SetColumnName (string ColumnName, int4 ColumnNumber)
```

Параметры

ColumnName	string	имя столбца
ColumnNumber	int4	номер столбца

Примеры



ПРИМЕР

Установить имя столбца.

```
Tree_1.SetColumnName("ID", 1);
```

8.13.3.10. SetColumnWidth

Устанавливает ширину столбца.

Синтаксис

```
void SetColumnWidth (int4 ColumnNumber, int4 w)
```

Параметры

ColumnNumber	int4	номер столбца
w	int4	ширина (в пикселях)

Примеры



ПРИМЕР

Установить ширину столбца.

```
Tree_1.SetColumnWidth(1, 300);
```

8.13.3.11. SetColumnHidden

Устанавливает видимость столбца.

Синтаксис

```
void SetColumnHidden (int4 ColumnNumber, bool f)
```

Параметры

ColumnNumber	int4	номер столбца
f	bool	флаг включения/отключения видимости столбца

Примеры



ПРИМЕР

Скрыть столбец.

```
Tree_1.SetColumnHidden(1, true);
```

8.13.3.12. SetItemData

Добавляет данные элементу в указанном столбце.

Синтаксис

```
void SetItemData (string ItemID, variant Data, int4 ColumnNumber)
```

Параметры

ItemID	string	идентификатор элемента
Data	variant	добавляемое элементу значение
ColumnNumber	int4	номер столбца

Примеры



ПРИМЕР

Добавить строковый идентификатор формы в элемент.

```
Tree_1.SetItemData(item_2, "f6d8d7a5-a2fc-438b-8e15-4858cada6df1", 0);
```

8.13.3.13. GetItemData

Возвращает данные (тип variant), хранящиеся в элементе указанного столбца.

Синтаксис

```
variant GetItemData (string ItemID, int4 ColumnNumber)
```

Параметры

ItemID	string	идентификатор элемента
--------	--------	------------------------

ColumnNumber	int4	номер столбца
--------------	------	---------------

Примеры



ПРИМЕР

Вывести данные элемента в текстовом поле.

```
itemdata = Tree_1.GetItemData(item_2, 0);  
TextEdit_2.Text = Variant.ToString(itemdata);
```

8.13.3.14. GetItemText

Возвращает строковое имя элемента указанного столбца.

Синтаксис

```
string GetItemText (string ItemID, int4 ColumnNumber)
```

Параметры

ItemID	string	идентификатор элемента
ColumnNumber	int4	номер столбца

Примеры



ПРИМЕР

Вывести имя элемента в текстовом поле.

```
itemtext = Tree_1.GetItemText(item_1, 0);  
TextEdit_2.Text = itemtext;
```

8.13.3.15. SetItemText

Задаёт имя элемента в указанном столбце.

Синтаксис

```
void SetItemText (string ItemID, string ItemText, int4 ColumnNumber)
```

Параметры

ItemID	string	идентификатор элемента
ItemText	string	имя элемента
ColumnNumber	int4	номер столбца

Примеры



ПРИМЕР

Задать имя элемента.

```
Name: string = "Третья форма";  
Tree_1.SetItemText(item_16, Name, 1);
```

8.13.3.16. SetItemBackgroundColor

Задаёт цвет фона элемента в указанном столбце.

Синтаксис

```
void SetItemBackgroundColor (string ItemID, uint4 color, int4 ColumnNumber)
```

Параметры

ItemID	string	идентификатор элемента
color	uint4	код цвета (в шестнадцатеричной или десятичной системе счисления)
ColumnNumber	int4	номер столбца

Примеры



ПРИМЕР

Задать цвет фона элемента.

```
Tree_1.SetItemBackgroundColor (item_2, 0x9CA2A3, 0);
```

8.13.3.17. SetItemTextColor

Задаёт цвет фона элемента в указанном столбце.

Синтаксис

```
void SetItemTextColor (string ItemID, uint4 color, int4 ColumnNumber)
```

Параметры

ItemID	string	идентификатор элемента
color	uint4	код цвета (в шестнадцатеричной или десятичной системе счисления)
ColumnNumber	int4	номер столбца

Примеры



ПРИМЕР

Задать цвет текста элемента.

```
Tree_1.SetItemTextColor (item_2, 0x9CA2A3, 0);
```

8.13.3.18. SetItemIconPath

Устанавливает иконку для элемента в указанном столбце.

Синтаксис

```
void SetItemIconPath (string ItemID, string pathToIcon, int4 ColumnNumber)
```

Параметры

ItemID	string	идентификатор элемента
pathToIcon	string	путь к файлу с иконкой относительно папки resources (вложенная папка проекта)
ColumnNumber	int4	номер столбца

Примеры



ПРИМЕР

Задать иконку элемента.

```
Tree_1.SetItemIconPath(item_2, "icons/alarms/n_3.png", 0);
```

9. Компоненты для проверки ввода и преобразования форматов

9.1. Входной формат

Преобразователь строковых значений (тип string) к универсальному типу variant. Преобразование выполняются в соответствии с форматом om.automation.controls.

9.1.1. Свойства

9.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

9.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

9.1.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

9.1.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

9.1.1.5. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

9.1.1.6. Ссылка на формат

Ссылка на экземпляр одного из типов `om.automation.controls`. Определяет формат для преобразования входного значения.

9.1.1.7. Входное значение

```
string InValue
```

Строка, которую нужно преобразовать к типу `variant`.

Примеры



ПРИМЕР

Подать на вход преобразователю значение сигнала (строку).

```
InputFormatter.InValue = OpcItem.Value;
```

9.1.1.8. Ошибка форматирования

```
string FormatErrorString
```

Свойство содержит текстовое описание ошибки преобразования.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать текст ошибки преобразования в текстовое поле.

```
TextEdit.Text = InputFormatter.FormatErrorString;
```

9.1.1.9. Признак ошибки форматирования

```
bool FormatError
```

Индикатор возникновения ошибки в процессе преобразования формата.

Значение

true	ошибка преобразования
false	преобразование прошло без ошибок



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

9.1.1.10. Выходное значение

```
variant OutValue
```

Значение, преобразованное к типу variant.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать преобразованное значение в поле ввода.

```
TextEdit.Text = InputFormatter.OutValue;
```

9.1.2. События

9.1.2.1. ValueChanged

Изменение входного значения (InValue) преобразователя формата.

Параметры

value	variant	значение, преобразованное к типу variant
-------	---------	------------------------------------------

9.1.2.2. FormatErrorChanged

Ошибка преобразования к какому-либо формату.

Параметры

hasError	bool	наличие ошибки форматирования: <ul style="list-style-type: none">➤ «true» - ошибка есть;➤ «false» - ошибки нет.
----------	------	----------------------------------------------------------------------------------------------------------------------------------------

9.2. Выходной формат

Преобразователь входного значения универсального типа variant к строковому типу (string). Преобразование выполняются в соответствии с форматом от.automation.controls.

9.2.1. Свойства

9.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

9.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиначный объект
>«1»	массив соответствующей размерности

9.2.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

9.2.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

9.2.1.5. Индекс

```
int8 Index
```

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

9.2.1.6. Ссылка на формат

Ссылка на экземпляр одного из типов `om.automation.controls`. Определяет формат для преобразования входного значения.

9.2.1.7. Входное значение

```
string InValue
```

Строка, которую нужно преобразовать к типу `string`.

Примеры

**ПРИМЕР**

Подать на вход преобразователю значение текстовой области.

```
OutputFormatter.InValue = TextEdit.Text;
```

9.2.1.8. Ошибка форматирования

```
string FormatErrorString
```

Свойство содержит текстовое описание ошибки преобразования.

**ПРИМЕЧАНИЕ**

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать текст ошибки преобразования в текстовое поле.

```
TextEdit.Text = OutputFormatter.FormatErrorString;
```

9.2.1.9. Признак ошибки форматирования

```
bool FormatError
```

Индикатор возникновения ошибки в процессе преобразования формата.

Значение

true	ошибка преобразования
false	преобразование прошло без ошибок



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

9.2.1.10. Выходное значение

```
variant OutValue
```

Значение, преобразованное к типу string.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать преобразованное значение в поле ввода.

```
TextEdit.Text = OutputFormatter.OutValue;
```

9.2.2. События

9.2.2.1. ValueChanged

Изменение значения на выходе преобразователя формата.

Параметры

value	variant	значение, преобразованное к типу string
-------	---------	-----------------------------------------

9.2.2.2. FormatErrorChanged

Ошибка преобразования к какому-либо формату.

Параметры

hasError	bool	наличие ошибки форматирования: <ul style="list-style-type: none">> «true» - ошибка есть;> «false» - ошибки нет.
----------	------	----------------------------------------------------------------------------------------------------------------------------------------------

9.3. Проверка ввода

Компонент для проверки различных типов данных на соответствие заданным условиям и ограничениям.

9.3.1. Свойства

9.3.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

9.3.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

9.3.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

9.3.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

9.3.1.5. Ссылка на тип проверки

ValidationType

Ссылка на экземпляр одного из типов-валидаторов `om.automation.controls`. Определяет формат и условия валидации.

9.3.1.6. Ввод

string Input

Строка, которую следует валидировать по указанным условиям.

Примеры

**ПРИМЕР**

Проверить адрес электронной почты.

```
InputValidator.Input = mymail@domain.com;
```

9.3.1.7. Результат проверки

UInt4 Result

Результат валидации в виде кода.

Значение

0	значение не валидировано
1	значение валидировано частично
2	значение валидировано полностью



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

9.3.2. События

9.3.2.1. ResultChanged

Смена кода результата валидации.

Параметры

result	int4	Результат валидации в виде кода: <ul style="list-style-type: none">> 0 - значение не валидировано;> 1- значение валидировано частично;> 2 - значение валидировано полностью.
--------	------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9.4. Проверка ввода регулярного выражения

Компонент для определения условий валидации по регулярному выражению.

9.4.1. Свойства

9.4.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

9.4.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

9.4.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

9.4.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

9.4.1.5. Частично верное выражение

```
string PartialValidPattern
```

Регулярное выражение, при удовлетворении которого результирующий код валидации (свойство Result компонента Проверка ввода) будет равен 1 (значение валидировано частично).

Примеры

**ПРИМЕР**

Записать в текстовое поле регулярное выражение для частичной валидации.

```
TextField.Text = ValidationRegex.PartialValidPattern = "[a-z0-9]+@[a-z0-9]+\\.\\w  
{2,6}$";
```

9.4.1.6. Верное выражение

```
string ValidPattern
```

Регулярное выражение, при удовлетворении которого результирующий код валидации (свойство Result компонента Проверка ввода) будет равен 2 (значение валидировано полностью).

Примеры

**ПРИМЕР**

Записать в текстовое поле регулярное выражение для полной валидации.

```
TextField.Text = ValidationRegex.PartialValidPattern = "[a-z0-9.-]+@[a-z0-9.-]+\\.\\w  
{2,6}$";
```

9.5. Проверка ввода целого числа

Компонент для определения условий валидации целочисленных значений.

9.5.1. Свойства

9.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

9.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

9.5.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

9.5.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

9.5.1.5. Нижняя граница

```
int4 Bottom
```

Минимальное значение, при котором возможна валидация.

9.5.1.6. Верхняя граница

```
int4 Top
```

Максимальное значение, при котором возможна валидация.

9.6. Проверка ввода числа с плавающей запятой

Компонент для определения условий валидации чисел с плавающей запятой.

9.6.1. Свойства

9.6.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

9.6.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

9.6.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

9.6.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

9.6.1.5. Число знаков после запятой

```
uint1 DigitsCount
```

Точность числа с плавающей запятой.

9.6.1.6. Нижняя граница

```
float Bottom
```

Минимальное значение, при котором возможна валидация.

9.6.1.7. Верхняя граница

float Top

Максимальное значение, при котором возможна валидация.

9.7. Формат булевского значения

Формат представления булевых значений.

9.7.1. Свойства

9.7.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

9.7.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

9.7.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

9.7.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```


9.7.1.5. Значение "Истина"

```
string TrueString
```

Строка, которая соответствует значению «True».

9.7.1.6. Значение "Ложь"

```
string FalseString
```

Строка, которая соответствует значению «False».

9.8. Формат даты/времени

Формат представления даты и времени.

9.8.1. Свойства

9.8.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

9.8.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

9.8.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

9.8.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

9.8.1.5. Формат даты/времени

```
string DateTimeFormat
```

Формат представления метки времени. Доступны обозначения: «dd» - дни, «MM»-месяцы, «уууу» - год, «Н» - часы, «mm» - минуты, «ss» - секунды.

9.8.1.6. Локальное время

```
bool LocalTime
```

Значение

true	оригинальная метка времени заменяется локальным временем
false	

9.9. Формат числа с плавающей запятой

Формат представления чисел с плавающей точкой.

9.9.1. Свойства

9.9.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

9.9.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

9.9.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

9.9.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

9.9.1.5. Число знаков после запятой

uint4 DigitsCount

Точность числа с плавающей запятой.

10. Компоненты периодического выполнения

10.1. Таймер

Компонент для циклического выполнения обработчика события **OnTimer**, которое повторяются с заданной периодичностью (свойство **Период**).

10.1.1. Свойства

10.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

10.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

10.1.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

10.1.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

10.1.1.5. Период

int4 Period

Период исполнения события по таймеру.

10.1.1.6. Активность

bool Active

Активность таймера.

Значение

true	событие по таймеру будет исполняться
false	событие по таймеру не будет исполняться

10.1.1.7. Текущее время

Now

Текущее локальное время.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

10.1.2. События


10.1.2.1. OnTimer

Событие, выполняемое по таймеру с заданным периодом.

Параметры

time	time stamp	текущее время UTC
------	------------	-------------------

Примеры



ПРИМЕР

Применить анимацию в виде меняющихся цветов для линейного градиента. Скорость анимации зависит от периодичности таймера.

```
Object.LinearGradient_1.GradientPoint_1.Color += 1;  
Object.LinearGradient_1.GradientPoint_2.Color -= 1;
```


10.2. Мигание

Компонент периодического выполнения, позволяющий графическим объектам мигать. Может применяться для мигания задвижек или других объектов технологического процесса.

10.2.1. Свойства

10.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

10.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

10.2.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

10.2.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

10.2.1.5. Период

uint4 Period

Период мигания графического объекта (мс).

10.2.1.6. Активность

bool Active

Активность мигания.

Значение

true	мигание включено
false	мигание выключено

10.2.1.7. Цвет пера

uint4 PenColor

Цвет внешней границы мигающего объекта. При мигании объекта цвет внешней границы меняется с цвета, указанного в свойствах объекта, на цвет, заданный в свойствах объекта **Мигание**.

10.2.1.8. Цвет заливки

uint4 BrushColor

Цвет внутренней заливки мигающего объекта. При мигании объекта цвет заливки меняется с цвета, указанного в свойствах объекта, на цвет, заданный в свойствах объекта **Мигание**.

10.2.1.9. Цвет шрифта

uint4 FontColor

Цвет шрифта мигающего объекта. При мигании объекта цвет шрифта меняется с цвета, указанного в свойствах объекта, на цвет, заданный в свойствах объекта типа **Мигание**.

10.2.2. События

10.2.2.1. Blink

Мигание объекта.

Параметры

value	bool	Отражает факт смены цвета пера/заливки/шрифта при мигании: <ul style="list-style-type: none">➤ true - цвет сменился на конечный;➤ false - цвет вернулся к начальному значению.
-------	------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

10.3. Команда

Компонент позволяет определять пользовательские команды, которые можно многократно вызывать из скриптов проекта. Компонент невидимый (не отображается на форме) и виден только в области **Структура объекта**. Компонент содержит процедуру, которая определяет логику команды. Процедура определяется в обработчике события **Invoked**. Чтобы выполнить команду, активируйте свойство **Триггер** или вызовите функцию **Invoke()**.



ПРИМЕЧАНИЕ

Если команда вызывается из скрипта, то команда будет выполнена после выполнения основного кода скрипта.

10.3.1. Свойства

10.3.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

10.3.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

10.3.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

10.3.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

10.3.1.5. Включено

bool Enabled

Включает/выключает пользовательскую команду.

Значение

true	команда включена
false	команда выключена

10.3.1.6. Триггер

bool InvokeTrigger

Свойство, запускающее процедуру в обработке **Invoked**.

Значение

true	запуск команды
false	остановка выполнения команды

10.3.2. События

10.3.2.1. Invoked

Активация свойства **Триггер** или вызова функции **Invoke()**.

10.3.2.2. EnabledChanged

Изменение свойства **Включено**.

Параметры

enabled	bool	состояние команды: <ul style="list-style-type: none">➤ true - команда включена;➤ false - команда выключена.
---------	------	------------------------------------------------------------------------------------------------------------------------------------

10.3.3. Функции

10.3.3.1. Invoke

Запускает пользовательскую команду, указанную в обработчике события **Invoked**.

Синтаксис

```
Invoke()
```

10.4. Звук

Компонент для проигрывания звуковых файлов. Звуковые файлы должны быть в формате *.wav и находиться в папке проекта resources.

10.4.1. Свойства

10.4.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

10.4.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

10.4.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

10.4.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```


**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

10.4.1.5. Число циклов

```
int4 LoopCount
```

Число повторений звукового файла.

10.4.1.6. Громкость

Громкость проигрывания звукового файла.

Значение

Может принимать значения от «0» до «1». «0» - отсутствие звука, «1» - максимальная громкость, совпадающая с громкостью динамиков.

Примеры

**ПРИМЕР**

Уменьшить громкость звука наполовину.

```
Sound.Volume = 0.5;
```

10.4.1.7. Путь до файла

```
string Path
```

Путь к звуковому файлу относительно папки resources.

10.4.1.8. Активность

```
bool Active
```

Начинает/прекращает проигрывание звукового файла.

Значение

true	звук проигрывается
false	звук не проигрывается

10.4.2. События

10.4.2.1. LoopPlayed

Завершение одного цикла проигрывания звука.

10.4.2.2. AllLoopsPlayed

Завершение всех циклов проигрывания звука.

10.5. Инициализатор

Компонент позволяет определить пользовательские команды, которые выполняются сразу при открытии экранной формы в рантайме. Компонент является невидимым (не отображается на форме) и виден только в области **Структура** объекта. Компонент состоит из события **Initialized**. В обработчике события указываются команды, которые должны выполняться сразу после запуска экранной формы.

Отличие компонента от компонента **Команда** в том, что **Команда** выполняется уже после запуска экранной формы (если использовать событие **Opened**), а событие **Инициализатора** выполнится сразу же с открытием экранной формы.

10.5.1. Свойства

10.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

10.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
-----	------------------

>«1»	массив соответствующей размерности
------	------------------------------------

10.5.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

10.5.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

10.5.2. События

10.5.2.1. Initialized

Открытие экранной формы в рантайме.

Примеры

**ПРИМЕР**

Активировать пользовательскую команду с запуском экранной формы.

```
Action.Enabled = true;
```

11. Компоненты для работы с графиками

11.1. График: данные

Компонент, выполняющий чтение оперативных или исторических данных с источника.

11.1.1. Свойства

11.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

11.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

11.1.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

11.1.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

11.1.1.5. Размер буфера данных

uint4 LengthReserved

Размер буфера данных (в количестве точек).

11.1.1.6. Загрузка из всех колонок данных

bool LoadColumns

Загрузка данных из всех колонок либо только из указанных.

Значение

true	загружать данные из всех колонок
false	загружать данные из колонок, указанных в компоненте График: линия

11.1.1.7. Сортировка

bool Sort

Разрешение/запрет сортировки данных.

Значение

true	сортировка разрешена
false	сортировка запрещена

11.1.1.8. Колонка для сортировки

uint4 SortColumnNumber

Номер колонки для сортировки.

11.1.1.9. Размер данных

uint4 Size

Максимальное количество точек, одновременно отображаемых на графике.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

11.1.2. События

11.1.2.1. ReadFinished

Завершение чтения данных.

Параметры

rowsRead	uint8	количество прочитанных строк
elapsedMs	float	время чтения данных (мсек)
errorCode	uint1	код ошибки, возникшей при чтении: <ul style="list-style-type: none">➤ 0 - загрузка завершена успешно;➤ 1 - отсутствует указанный файл;➤ 2 - недостаточно прав для чтения файла.
message	string	сообщение

11.1.2.2. SaveFinished

Завершение сохранения данных.

Параметры

rowsSave	uint8	количество сохраненных строк
errorCode	uint1	код ошибки, возникшей при сохранении: <ul style="list-style-type: none">➤ 0 - запись завершена успешно;➤ 1 - недостаточно прав для записи файла.

11.1.3. Функции

11.1.3.1. BeginReadAsync

Обновляет данные через IRowSet.

Синтаксис

```
void BeginReadAsync()
```


11.1.3.2. CancelReadAsync

Отменяет обновление данных через IRowSet.

Синтаксис

```
void CancelReadAsync()
```

11.1.3.3. EnableRT

Разрешает/запрещает чтение данных в режиме реалтайма.

Синтаксис

```
void EnableRT(EnableFlag)
```

Параметры

EnableFlag	bool	флаг разрешения/запрещения чтения данных
------------	------	------------------------------------------

11.1.3.4. Clear

Очищает содержимое.

Синтаксис

```
void Clear()
```

11.1.3.5. Save

Сохраняет данные в файл в формате CSV или JSON.

Синтаксис

```
void Save(FilePath, Format)
```

Параметры

FilePath	string	полный путь к файлу
Format	string	формат данных

11.1.3.6. Load

Загружает данные из файла в формате CSV или JSON.

Синтаксис

```
void Load(FilePath, Format)
```

Параметры

FilePath	string	полный путь к файлу
Format	string	формат данных

11.2. График: декоратор линии

Компонент позволяет декорировать линию.

11.2.1. Свойства

11.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

11.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

11.2.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

11.2.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

11.2.1.5. Цвет

uint4 Color

Цвет элемента.

11.2.1.6. Название

string Title

Название элемента.

11.2.1.7. Толщина линии

float LineWidth

Толщина линии.

11.2.1.8. Размер точки

uint4 PointSize

Размер точки пересечения с элементом.

11.2.1.9. Стилль линии

uint4 LineBrushStyle

Стилль линии элемента.

11.2.1.10. Активность

```
bool Active
```

Активирует/деактивирует декоратор.

Значение

true	декоратор активен
false	декоратор неактивен

11.2.1.11. Значение минимума декоратора

```
float MinValue
```

Значение минимума декоратора.

11.2.1.12. Значение максимума декоратора

```
float MaxValue
```

Значение максимума декоратора.

11.2.1.13. Номер колонки декоратора

```
int4 ColumnNumber
```

Номер колонки данных в источнике, предоставляющем данные для определения условий декорирования.

11.2.1.14. Приоритет

```
uint4 zOrder
```

Приоритет элемента.

11.3. График: линия

Компонент позволяет отобразить линию на основе данных, полученных от источника.

11.3.1. Свойства

11.3.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

11.3.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

11.3.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

11.3.1.4. Индекс

```
int8 Index
```

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

11.3.1.5. Цвет

```
uint4 Color
```

Цвет элемента.

11.3.1.6. Название

```
string Title
```

Название элемента.

11.3.1.7. Единица измерения

```
string Unit
```

Единицы измерения для линии.

11.3.1.8. Толщина линии

```
float LineWidth
```

Толщина линии.

11.3.1.9. Размер точки

```
uint4 PointSize
```

Размер точки пересечения с элементом.

11.3.1.10. Стилъ линии

```
uint4 LineBrushStyle
```

Стилъ линии элемента.

11.3.1.11. Прозрачность

```
float BrushOpacity
```

Прозрачность элемента.

11.3.1.12. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
------	--------------

false	объект скрыт
-------	--------------

11.3.1.13. Приоритет

uint4 zOrder

Приоритет элемента.

11.3.1.14. Тип графика

uint4 DrawingType

Тип графика.

Значение

0	линейный график
1	ступенчатый график
2	гистограмма

11.3.1.15. Номер колонки значений X

uint4 ColumnNumberTime

Номер колонки для временных значений.

11.3.1.16. Номер колонки значений Y

uint4 ColumnNumberValue

Задаёт номер колонки для числовых значений.

11.3.1.17. Вывод значений пересечений с реперной линией

bool ShowActiveControlLineLables

Скрывает/отображает значения пересечений с реперной линией.

Значение

true	отображает значения пересечений
false	скрывает значения пересечений

11.3.1.18. Идентификатор линии

uint4 ID

Идентификатор (ID) линии.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

11.3.2. События

11.3.2.1. ReadFinished

Завершение чтения данных.

Параметры

rowsRead	uint8	количество прочитанных строк
elapsedMs	float	время чтения данных (мсек)
errorCode	uint1	код ошибки, возникшей при чтении: <ul style="list-style-type: none"> > 0 - загрузка завершена успешно; > 1 - отсутствует указанный файл; > 2 - недостаточно прав для чтения файла.
message	string	сообщение

11.3.2.2. SaveFinished

Завершение сохранения данных.

Параметры

rowsSave	uint8	количество сохраненных строк
----------	-------	------------------------------

errorCode	uint1	код ошибки, возникшей при сохранении: <ul style="list-style-type: none">➤ 0 - запись завершена успешно;➤ 1 - недостаточно прав для записи файла.
-----------	-------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

11.3.3. Функции

11.3.3.1. SetColor

Задаёт цвет линии.

Синтаксис

```
void SetColor(Color)
```

Параметры

color	uint4	код цвета линии
-------	-------	-----------------

11.3.3.2. GetActiveControlLineLables

Возвращает значения пересечений с активной реперной линией в виде строки.

Синтаксис

```
string GetActiveControlLineLables()
```

11.3.3.3. GetMinX

Возвращает минимум по оси X (тип variant).

Синтаксис

```
variant GetMinX()
```

11.3.3.4. GetMaxX

Возвращает максимум по оси X (тип variant).

Синтаксис

```
variant GetMaxX()
```

11.3.3.5. GetMinY

Возвращает минимум по оси Y (тип variant).

Синтаксис

```
variant GetMinY()
```

11.3.3.6. GetMaxY

Возвращает максимум по оси Y (тип variant).

Синтаксис

```
variant GetMaxY()
```

11.3.3.7. GetGlobalMinX

Возвращает минимум загруженных данных по оси X (тип variant).

Синтаксис

```
variant GetGlobalMinX()
```

11.3.3.8. GetGlobalMaxX

Возвращает максимум загруженных данных по оси X (тип variant).

Синтаксис

```
variant GetGlobalMaxX()
```

11.3.3.9. GetGlobalMinY

Возвращает минимум загруженных данных по оси Y (тип variant).

Синтаксис

```
variant GetGlobalMinY()
```

11.3.3.10. GetGlobalMaxY

Возвращает максимум загруженных данных по оси Y (тип variant).

Синтаксис

```
variant GetGlobalMaxY()
```

11.3.3.11. SetAxisByld

Назначает ось по ID.

Синтаксис

```
void SetAxisByld
```

Параметры

AxisID	int4	идентификатор (ID) оси
--------	------	------------------------

11.3.3.12. Save

Сохраняет данные в файл в формате CSV или JSON.

Синтаксис

```
void Save(FilePath, Format)
```

Параметры

FilePath	string	полный путь к файлу
Format	string	формат данных

11.3.3.13. Load

Загружает данные из файла в формате CSV или JSON.

Синтаксис

```
void Load(FilePath, Format)
```

Параметры

FilePath	string	полный путь к файлу
Format	string	формат данных

11.4. График: ось

Компонент позволяет добавить на трендовое поле оси для отображения графиков в определенном масштабе.

11.4.1. Свойства

11.4.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

11.4.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

11.4.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

11.4.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

11.4.1.5. Цвет

uint4 Color

Цвет элемента.

11.4.1.6. Ориентация оси

uint4 Type

Значение

1	горизонтальная
2	вертикальная

11.4.1.7. Выравнивание

uint4 Align

Положение оси относительно поля графика.

Значение

1	слева для вертикальной оси, снизу для горизонтальной
2	справа для вертикальной оси, сверху для горизонтальной

11.4.1.8. Тип данных

uint4 DataType

Тип данных оси.

Значение

0	числовые значения
2	время

11.4.1.9. Количество делений

```
uint4 Ticks
```

Количество делений на оси.

11.4.1.10. Тип масштабирования

```
uint4 Scale
```

Тип масштабирования оси.

Значение

0	автоматическое масштабирование отключено
1	полное автоматическое масштабирование. Размер оси может изменяться в зависимости от отображаемых данных: как в сторону расширения, так и в сторону уменьшения
2	автоматическое расширение масштаба. Размер оси может изменяться только в сторону увеличения

11.4.1.11. Логарифмическая шкала

```
bool ScaleLog
```

Скрытие/отображение логарифмической шкалы (деления на шкале проставлены в соответствии со степенями числа 10).

Значение

true	шкала отображается
false	шкала скрыта

11.4.1.12. Единицы измерения

```
string Unit
```

Единицы измерения отображаемых данных.

11.4.1.13. Положение единиц измерения

uint4 UnitPos

Положение единиц измерения на оси.

Значение

0	слева для горизонтальной оси, снизу для вертикальной
1	по центру
2	справа для горизонтальной оси, сверху для вертикальной

11.4.1.14. Формат меток

string Format

Формат меток на оси.

11.4.1.15. Формат метки времени

string FormatT

Формат меток времени на оси.

11.4.1.16. Минимум

variant Min

Минимальное значение отображаемых данных.

11.4.1.17. Максимум

variant Max

Максимальное значение отображаемых данных.

11.4.1.18. Название оси

```
string AxisName
```

Название, отображаемое рядом с осью.

11.4.1.19. Показывать единицы измерения

```
bool ShowUnit
```

Скрытие/отображение единиц измерения.

Значение

true	единицы измерения видны
false	единицы измерения скрыты

11.4.1.20. Положение названия оси

```
uint4 AxisNamePos
```

Положение названия оси.

Значение

0	слева для горизонтальной оси, снизу для вертикальной
1	по центру
2	справа для горизонтальной оси, сверху для вертикальной

11.4.1.21. Видимость оси

```
bool ShowUnit
```

Скрытие/отображение оси.

Значение

true	ось видна
false	ось скрыта

11.4.1.22. Шрифт

```
string Font
```

Шрифт элемента.

11.4.1.23. Коэффициент масштабирования

```
uint4 Zoom
```

Коэффициент масштабирования оси в процентах.

11.4.2. События

11.4.2.1. OnRangeChange

Изменение диапазона выводимых данных.

Параметры

Min	float	минимальное значение диапазона
Max	float	максимальное значение диапазона

11.4.2.2. OnClickValue

Выбор значения репера щелчком мыши.

Параметры

clickValue	variant	координата курсора по оси, соответствующая месту щелчка мыши
------------	---------	--------------------------------------------------------------

11.4.2.3. OnWheelMouse

Вращение колеса мыши.

Параметры

steps	float	шаги поворота (знак указывает направление)
left	float	координата курсора по оси X
top	float	координата курсора по оси Y

keyboardmodifier	int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
------------------	------	-----------------------------------------------------------------------

11.4.3. Функции

11.4.3.1. setAxisType

Устанавливает ориентацию оси.

Синтаксис

```
void setAxisType(Type)
```

Параметры

Type	uint4	значения: <ul style="list-style-type: none"> > «1» - горизонтальная; > «2» - вертикальная.
------	-------	-----------------------------------------------------------------------------------------------------------------------------

11.4.3.2. SetInterval

Устанавливает интервал оси.

Синтаксис

```
void SetInterval (MinValue, MaxValue)
```

Параметры

MinValue	variant	минимальное значение
MaxValue	variant	максимальное значение

11.4.3.3. GetID

```
GetID()
```

Возвращает ID оси.

11.5. График: поле графика

Компонент позволяет подготовить область для отображения графиков.

11.5.1. Свойства

11.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

11.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

11.5.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

11.5.1.4. Индекс

```
int8 Index
```

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

11.5.1.5. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры

**ПРИМЕР**

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

11.5.1.6. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

11.5.1.7. Видимость

bool Visible

Видимость объекта на форме.

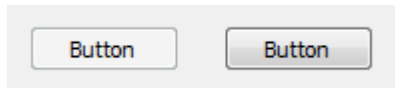
Значение

true	объект виден
false	объект скрыт

11.5.1.8. Включено

bool Enabled

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

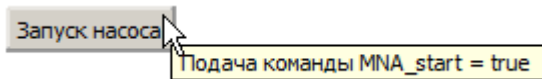
Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

11.5.1.9. Всплывающая подсказка

string Tooltip

Подсказка, всплывающая при наведении курсора на объект.



Примеры



ПРИМЕР

Задать всплывающую подсказку для объекта.

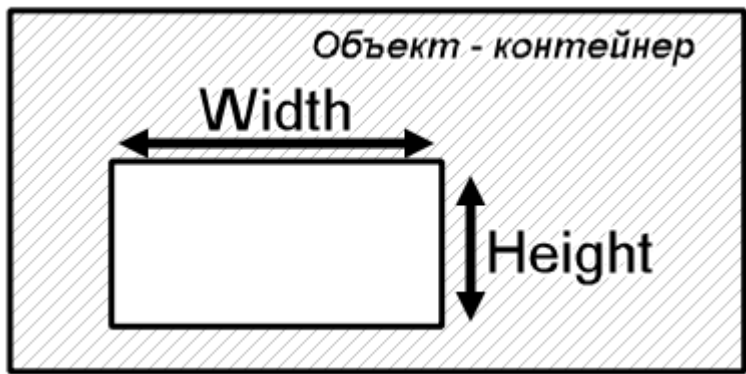
```
Object.Tooltip = "Подача команды MNA_start = true";
```

11.5.1.10. Ширина и Высота

float Width

float Height

Ширина и высота объекта в пикселях.



11.5.1.11. Размер точки курсора

uint4 CursorPointSize

Размер точки курсора на графике (в пикселях).

11.5.1.12. Толщина линии курсора

uint4 CursorLineWidth

Толщина линии курсора на графике.

11.5.1.13. Режим курсора

uint4 CursorMode

Режим курсора.

Значение

0	перемещение оси зажатой правой кнопкой мыши (значение по умолчанию)
1	масштабирование выделенной области

2	перемещение графика в нужном направлении
3	добавление, выделение, перемещение реперов

11.5.1.14. Цвет фона

```
uint4 BrushColor
```

Цвет фона графика.

11.5.1.15. Цвет курсора

```
uint4 CursorColor
```

Цвет курсора на графике.

11.5.1.16. Цвет активной реперной линии

```
uint4 ActiveControllLineColor
```

Цвет активной реперной линии графика.

11.5.1.17. Толщина активной реперной линии

```
uint4 ActiveControllLineWidth
```

Толщина активной реперной линии графика.

11.5.1.18. Стилль линии курсора

```
uint4 CursorLineBrushStyle
```

Стилль линии курсора на графике.

11.5.1.19. Таймлайн

```
bool ShowTimeLine
```

Скрытие/отображение панели таймлайна

Значение

true	панель таймлайна отображается
false	панель таймлайна скрыта

11.5.1.20. Легенда

bool ShowLegend

Скрытие/отображение легенды на графике.

Значение

true	легенда отображается
false	легенда скрыта

11.5.1.21. Выравнивание легенды

uint4 AlignLegend

Положение поля легенды на графике.

11.5.1.22. Экстраполяция

bool Extrapolation

Разрешение/запрет продолжения линии до конца выбранного интервала.

Значение

true	экстраполяция включена
false	экстраполяция выключена

11.5.1.23. Перемещать выноски репера

bool MovePointLables

Разрешение/запрет перемещения выносок репера.

Значение

true	разрешено перемещать выноски
false	запрещено перемещать выноски

11.5.1.24. Шрифт

```
string Font
```

Шрифт элемента.

11.5.1.25. Коэффициент масштабирования по X, %

```
uint4 ZoomX
```

Коэффициент масштабирования графика по оси X в процентах.

11.5.1.26. Коэффициент масштабирования по Y, %

```
uint4 ZoomY
```

Коэффициент масштабирования графика по оси Y в процентах.

11.5.1.27. Управление выравниванием

```
bool EnableScaleAxisX
```

Включение/отключение автоматического выравнивания по оси X при приближении ползунков таймлайна к предельным положениям.

Значение

true	автоматическое выравнивание включено
false	автоматическое выравнивание выключено

11.5.1.28. Автообновление

```
bool AutoUpdate
```

Разрешение/запрет автоматического обновления графика при изменении данных.

Значение

true	автоматическое обновление графика разрешено
false	автоматическое обновление графика запрещено

11.5.2. События

11.5.2.1. OnWheelMouse

Вращение колеса мыши.

Параметры

steps	float	шаги поворота (знак указывает направление)
left	float	координата курсора по оси X
top	float	координата курсора по оси Y
keyboardmodifier	int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

11.5.3. Функции

11.5.3.1. Update

Обновляет (перерисовывает) график.

Синтаксис

```
Update()
```

11.5.3.2. ResetZoom

Восстанавливает исходный масштаб графика.

Синтаксис

```
ResetZoom()
```

11.6. График: реперная линия

Компонент позволяет просматривать значения графика в точках пересечения линии с графиком сигнала.

11.6.1. Свойства

11.6.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

11.6.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

11.6.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

11.6.1.4. Индекс

```
int8 Index
```

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

11.6.1.5. Цвет

```
uint4 Color
```

Цвет элемента.

11.6.1.6. Толщина линии

```
float LineWidth
```

Толщина линии.

11.6.1.7. Значение

variant Value

Значение графика в точке пересечения

11.6.1.8. Стил

uint4 BrushStyle

Стил элемента.

11.6.1.9. Прозрачность

float BrushOpacity

Прозрачность элемента.

11.6.1.10. Отображение значений

bool ShowValues

Скрытие/отображение точек пересечения реперной линии с графиками.

Значение

true	точки пересечения отображаются
false	точки пересечения скрыты

11.6.1.11. Цвет фона значений

uint4 BgValuesColor

Цвет фона значений.

11.6.1.12. Прозрачность фона значений

float BgValuesOpacity

Прозрачность фона значений.

11.6.1.13. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

11.6.1.14. Формат меток на пересечении с линиями

```
string Format
```

Формат меток на пересечении с линиями.

11.6.1.15. Шрифт

```
string Font
```

Шрифт элемента.

11.6.1.16. Режим

```
uint4 Mode
```

Вид отображаемых значений.

Значение

0	точное значение
1	интервальные значения

11.6.2. События

11.6.2.1. ChangeValue

Изменение значения.

11.6.3. Функции

11.6.3.1. getActive

Возвращает активность репера.

Синтаксис

```
bool getActive()
```

11.6.3.2. SetActive

Устанавливает активность репера.

Синтаксис

```
void SetActive(ActiveFlag)
```

Параметры

ActiveFlag	bool	возможные значения: <ul style="list-style-type: none">➤ «true» - репер активен➤ «false» - репер неактивен
------------	------	----------------------------------------------------------------------------------------------------------------------------------

11.7. График: точка пересечения

Компонент позволяет настроить отображение данных в точке пересечения реперной линии с линией графика.

11.7.1. Свойства

11.7.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

11.7.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

11.7.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

11.7.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

11.7.1.5. Цвет

```
uint4 Color
```

Цвет элемента.

11.7.1.6. Стил

```
uint4 BrushStyle
```

Стил элемента.

11.7.1.7. Прозрачность

```
float BrushOpacity
```

Прозрачность элемента.

11.7.1.8. Цвет фона значений

```
uint4 BgValuesColor
```

Цвет фона значений.

11.7.1.9. Прозрачность фона значений

```
float BgValuesOpacity
```

Прозрачность фона значений.

11.7.1.10. Видимость

```
bool Visible
```

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

11.7.1.11. Шрифт

```
string Font
```

Шрифт элемента.

11.7.1.12. Текст

```
string ToolTipText
```

Текст всплывающей подсказки.

11.7.1.13. Признак пересечения с линией

```
bool Exists
```

Наличие пересечения с линией.

Значение

true	пересекает линию
false	не пересекает линию



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

11.7.1.14. Значение X

variant X

Значение точки пересечения по оси X.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

11.7.1.15. Значение Y

variant Y

Значение точки пересечения по оси Y.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

11.7.2. Функции

11.7.2.1. SetLineID

Задаёт значение ID линии.

Синтаксис

```
void SetLineID (ID)
```

Параметры

ID	uint4	идентификатор (ID) линии
----	-------	--------------------------

11.7.2.2. GetLineID

Возвращает ID линии.

Синтаксис

```
uint4 GetLineID()
```

11.7.2.3. FirstLine

Выбирает первую линию.

Синтаксис

```
bool FirstLine()
```

11.7.2.4. NextLine

Переходит к следующей линии.

Синтаксис

```
bool NextLine()
```

11.7.2.5. PrevLine

Переходит к предыдущей линии.

Синтаксис

```
bool PrevLine()
```

11.7.2.6. First

Выбирает первую точку.

Синтаксис

```
bool First()
```

11.7.2.7. Next

Переходит к следующей точке.

Синтаксис

```
bool Next()
```

11.7.2.8. Prev

Переходит к предыдущей точке.

Синтаксис

```
bool Prev()
```

11.7.2.9. GetValue

Возвращает значение из указанной колонки.

Синтаксис

```
GetValue(ColumnId)
```

Параметры

ColumnId	int4	идентификатор (ID) колонки
----------	------	----------------------------

11.8. График: экспорт

Компонент позволяет сохранять в файл данные, отображаемые на трендовом поле.

11.8.1. Свойства

11.8.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

11.8.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

11.8.1.3. Сортировка

```
bool Sort
```

Разрешение/запрет сортировки данных.

11.8.1.4. Колонка для сортировки

```
uint4 SortColumnNumber
```

Номер колонки для сортировки.

11.8.2. Функции

11.8.2.1. Reload

Запрашивает данные графика.

Синтаксис

```
void Reload()
```

12. Компоненты для формирования таблиц

12.1. Таблица

Компонент позволяет графически представить данные, хранящиеся в таблице модели данных.

12.1.1. Свойства

12.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

12.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

12.1.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

12.1.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

12.1.1.5. Модель данных

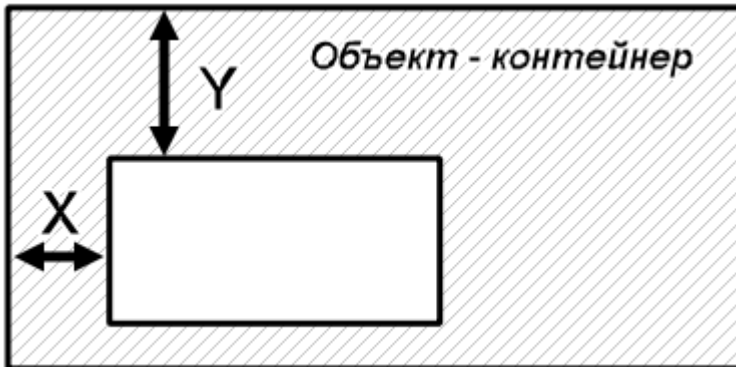
Ссылка на модель данных.

12.1.1.6. X и Y

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

12.1.1.7. Угол поворота

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

12.1.1.8. Масштаб

Float Scale

Коэффициент масштабирования объекта.

Примеры



ПРИМЕР

Установить полуторный масштаб объекта.

```
Object.Scale = 1.5;
```



ПРИМЕР

Увеличить текущий масштаб объекта в 3 раза.

```
Object.Scale *= 3;
```



ПРИМЕР

Уменьшить текущий масштаб объекта в 2 раза.

```
Object.Scale *= 0.5;
```

12.1.1.9. Видимость

bool Visible

Видимость объекта на форме.

Значение

true	объект виден
false	объект скрыт

12.1.1.10. Включено

```
bool Enabled
```

Активность объекта.



Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

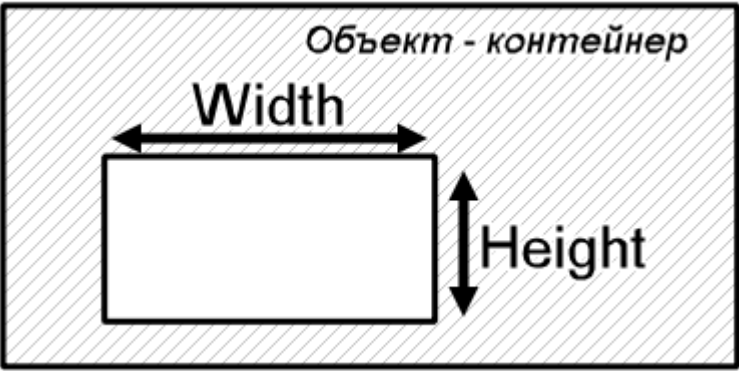
```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

12.1.1.11. Ширина и Высота

```
float Width
```

```
float Height
```

Ширина и высота объекта в пикселях.



12.1.1.12. Высота заголовка

```
int4 HeaderHeight
```

Высота заголовка таблицы (в пикселях). Может превышать указанное значение, если разрешен перенос слов.

12.1.1.13. Видимость заголовка

```
bool HeaderVisible
```

Скрывает/отображает заголовок таблицы.

Значение

true	заголовок отображается
false	заголовок скрыт

12.1.1.14. Высота строки

```
int4 RowHeight
```

Высота строки таблицы (в пикселях). Может превышать указанное значение, если разрешен перенос слов.

12.1.1.15. Порядок вывода строк

```
uint1 RowOrder
```

Расположение новых строк в таблице.

Значение

0	каждая новая строка добавляется в конец таблицы
1	каждая новая строка добавляется в начало таблицы

12.1.1.16. Поведение выделения

uint1 SelectionBehavior

Количество выделенных ячеек при нажатии на одну из них.

Значение

«0»	выделяется только ячейка
«1»	выделяется строка
«3»	выделяется столбец

12.1.1.17. Режим выделения

uint1 SelectionMode

Комбинация клавиш для выделения нескольких ячеек.

Значение

«0»	выделение ячеек отключено
«1»	возможность одновременного выделения только одной ячейки кликом мыши
«2»	возможность одновременного выделения нескольких ячеек и индивидуального снятия выделения
«3»	возможность выделения смежных элементов кликом мыши и клавишей «Ctrl»
«4»	возможность выделения смежных элементов кликом мыши и клавишей «Shift»

12.1.1.18. Следить за новой строкой

bool LatestRowTracking

Включение/отключение постоянного отображения в поле видимости последней добавленной строки.

Значение

0	режим слежения включен
1	режим слежения выключен

12.1.2. События

12.1.2.1. MousePress

Нажатие кнопки мыши в области строки или столбца.

Параметры

row	int4	порядковый номер строки. Нумерация начинается с 0
column	int4	порядковый номер столбца. Нумерация начинается с 0

12.1.2.2. MouseRelease

Нажатая кнопка мыши отпущена в области строки или столбца.

Параметры

row	int4	порядковый номер строки. Нумерация начинается с 0
column	int4	порядковый номер столбца. Нумерация начинается с 0

12.1.2.3. CurrentChanged

Выделение ячейки.

Параметры

row	int4	порядковый номер строки. Нумерация начинается с 0
column	int4	порядковый номер столбца. Нумерация начинается с 0

12.1.2.4. SelectionChanged

Изменение выделения строк.

12.1.3. Функции

12.1.3.1. GetColumnCount

Возвращает количество столбцов, существующих в таблице на момент вызова функции.

Синтаксис

```
int4 GetColumnCount()
```

12.1.3.2. GetRowCount

Возвращает количество строк, существующих в таблице на момент вызова функции.

Синтаксис

```
int4 GetRowCount()
```

12.1.3.3. GetRowDataAsJson

Получить JSON-объект с данными указанной строки.

Синтаксис

```
string GetRowDataAsJson(Row)
```

Параметры

Row	int	порядковый номер строки. Нумерация начинается с 0
-----	-----	---------------------------------------------------

12.1.3.4. GetCellData

Возвращает значение указанной ячейки.

Синтаксис

```
string GetCellData(Row, Column)
```

Параметры

Row	int	порядковый номер строки. Нумерация начинается с 0
Row	int	порядковый номер столбца. Нумерация начинается с 0

12.1.3.5. GetSelectedRows

Возвращает JSON-объект с номерами выделенных строк.

Синтаксис

```
string GetSelectedRows()
```

12.1.3.6. SetSelectedRows

Задать JSON-объект с номерами выделенных строк.

Синтаксис

```
void SetSelectedRows(Rows)
```

Параметры

Rows	string	JSON-объект
------	--------	-------------

12.1.3.7. AddRow

Добавляет новую строку в конец таблицы.

Синтаксис

```
void AddRow()
```

12.1.3.8. InsertRow

Добавляет новую строку после указанной строки в таблице.

Синтаксис

```
void InsertRow(Row)
```

Параметры

Row	int	порядковый номер строки, после которой будет вставлена новая строка. Нумерация начинается с 0
-----	-----	-----------------------------------------------------------------------------------------------

12.1.3.9. RemoveRow

Удаляет указанную строку из таблицы.

Синтаксис

```
void RemoveRow(Row)
```

Параметры

Row	int	порядковый номер строки, после которой будет вставлена новая строка. Нумерация начинается с 0
-----	-----	-----------------------------------------------------------------------------------------------

12.1.3.10. IsRowSelected

Возвращает true, если указанная строка выделена.

Синтаксис

```
bool IsRowSelected(Row)
```

Параметры

Row	int	порядковый номер строки, после которой будет вставлена новая строка. Нумерация начинается с 0
-----	-----	-----------------------------------------------------------------------------------------------

12.1.3.11. MoveColumn

Перемещает столбец из одной позиции в указанную позицию.

Синтаксис

```
void MoveColumn(Column, Position)
```

Параметры

Column	int	порядковый номер столбца. Нумерация начинается с 0
Position	int	номер позиции, в которую будет перемещен столбец

12.1.3.12. ApplySort

Применяет сортировку, настроенную в столбцах.

Синтаксис

```
void ApplySort()
```

12.1.3.13. MapColumnFromModel

Возвращает соответствующий индекс столбца из модели данных в таблице.

Синтаксис

```
int4 MapColumnFromModel(Column)
```

Параметры

Column	int4	Порядковый номер столбца. Нумерация начинается с 0.
--------	------	-----------------------------------------------------

12.1.3.14. MapColumnToModel

Возвращает соответствующий индекс столбца из таблицы в модели данных.

Синтаксис

```
int4 MapColumnToModel(Column)
```

Параметры

Column	int4	Порядковый номер столбца. Нумерация начинается с 0.
--------	------	-----------------------------------------------------

12.1.3.15. MapRowFromModel

Возвращает номер строки в видимой таблице по указанному номеру строки в модели.

Синтаксис

```
int4 MapRowFromModel(Row)
```

Параметры

Row	int4	Порядковый номер строки в модели. Нумерация начинается с 0.
-----	------	-------------------------------------------------------------

12.1.3.16. MapRowToModel

Возвращает номер строки в модели по указанному номеру строки видимой таблицы.

Синтаксис

```
int4 MapRowToModel(Row)
```

Параметры

Row	int4	Порядковый номер строки в видимой таблице. Нумерация начинается с 0.
-----	------	----------------------------------------------------------------------

12.1.3.17. ScrollToBottom

Прокручивает таблицу к концу.

Синтаксис

```
void ScrollToBottom()
```

12.1.3.18. ScrollToTop

Прокручивает таблицу к началу.

Синтаксис

```
void ScrollToTop()
```

12.1.3.19. ScrollTo

Прокручивает таблицу к указанной ячейке.

Синтаксис

```
void ScrollTo(Row, Column, Hint)
```

Параметры

Row	int	порядковый номер строки. Нумерация начинается с 0
Column	int	порядковый номер столбца. Нумерация начинается с 0
Hint	uint	месторасположение ячейки в видимой области после смещения. Возможные значения: <ul style="list-style-type: none"> > 0 - ячейка должна быть видима; > 1 - ячейка расположена сверху; > 2 - ячейка расположена снизу; > 3 - ячейка расположена по центру.

12.1.3.20. GetPageHeaderTextAlignment

Прокручивает таблицу к началу.

Синтаксис

```
int4 GetPageHeaderTextAlignment()
```

Возвращает выравнивание текста верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

12.1.3.21. SetPageHeaderTextAlignment

Задаёт выравнивание текста верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
void SetPageHeaderTextAlignment(Alignment)
```


Параметры

Alignment	string	Возможные значения: <ul style="list-style-type: none">➤ 1 - по левому краю;➤ 2 - по правому краю;➤ 4 - по центру.
-----------	--------	---------------------------------------------------------------------------------------------------------------------------------------------------

12.1.3.22. GetPageHeaderFont

Возвращает шрифт текста верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
string GetPageHeaderFont()
```

12.1.3.23. SetPageHeaderFont

Задаёт шрифт текста верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
int4 SetPageHeaderFont(Font)
```

Параметры

Font	string	наименование шрифта
------	--------	---------------------

12.1.3.24. GetPageHeaderText

Возвращает текст верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
string GetPageHeaderText()
```

12.1.3.25. SetPageHeaderText

Записывает текст в верхний колонтитул. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
void SetPageHeaderText(Text)
```

Параметры

Text	string	текст
------	--------	-------

12.1.3.26. GetPageFooterTextAlignment

Возвращает выравнивание текста нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
int4 GetPageFooterTextAlignment()
```

12.1.3.27. SetPageFooterTextAlignment

Задаёт выравнивание текста нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
void SetPageFooterTextAlignment(Alignment)
```

Параметры

Alignment	string	возможные значения: <ul style="list-style-type: none">➤ 1 - по левому краю;➤ 2 - по правому краю;➤ 4 - по центру.
-----------	--------	---------------------------------------------------------------------------------------------------------------------------------------------------

12.1.3.28. GetPageFooterFont

Возвращает шрифт текста нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
string GetPageFooterFont()
```

12.1.3.29. SetPageFooterFont

Задаёт шрифт текста нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
void SetPageFooterFont(Font)
```

Параметры

Font	string	наименование шрифта
------	--------	---------------------

12.1.3.30. GetPageFooterText

Возвращает текст нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
string GetPageFooterText()
```

12.1.3.31. SetPageFooterFont

Записать текст в нижний колонтитул. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
void SetPageFooterText(Text)
```

Параметры

Text	string	текст для нижнего колонтитула
------	--------	-------------------------------

12.1.3.32. GetPageFooterText

Возвращает текст нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
string GetPageFooterText()
```

12.1.3.33. SetPageFooterText

Записать текст в нижний колонтитул. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
string SetPageFooterText(Text)
```

Параметры

Text	string	текст для нижнего колонтитула
------	--------	-------------------------------

12.1.3.34. GetPageNumberFont

Возвращает шрифт номера страницы. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
string GetPageNumberFont()
```

12.1.3.35. SetPageNumberFont

Задаёт шрифт номера страницы. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.

Синтаксис

```
void SetPageNumberFont(Font)
```

Параметры

Font	string	наименование шрифта
------	--------	---------------------

12.1.3.36. Print

Запускает печать таблицы.

Синтаксис

```
void Print()
```

12.1.3.37. PrintDialog

Открывает диалог печати.

Синтаксис

```
void PrintDialog()
```

12.1.3.38. Save

Сохраняет таблицу в файл.

Синтаксис

```
void Save(FileName, Format)
```

Параметры

FileName	string	имя сохраняемого файла
Format	string	формат сохраняемого файла

12.2. Таблица: модель данных

Компонент позволяет сформировать невизуальную таблицу, предназначенную для хранения данных. Данные могут поступать в таблицу от источника AP либо вводиться вручную с помощью скриптов

SePlatform.Om или JavaScript.

12.2.1. Свойства

12.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

12.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

12.2.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

12.2.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

12.2.1.5. Заморозка

```
bool Frozen
```

Запрещает/разрешает обновление таблицы данными, поступившими от источника или заданными в скрипте.

12.2.1.6. Максимальное количество рядов данных

```
int4 MaxRowCount
```

Максимальное число строк таблицы.

12.2.2. События

12.2.2.1. DataChanged

Изменение содержимого в таблице модели данных.

Параметры

top	int4	порядковый номер первой строки. в которой были изменения. Нумерация начинается с 0
left	int4	порядковый номер столбца в первой строке, в котором были изменения. Нумерация начинается с 0
bottom	int4	порядковый номер последней строки. в которой были изменения. Нумерация начинается с 0
right	int4	порядковый номер столбца в последней строке, в котором были изменения. Нумерация начинается с 0

12.2.2.2. ReadFinished

Завершение чтения данных моделью.

Параметры

rowsRead	unt8	количество прочитанных строк
elapsedSec	float	общее время чтения (в секундах)

12.2.2.3. RowsInserted

Вставка строки.

Параметры

first	int4	индекс первой вставленной строки
last	int4	индекс последней вставленной строки

12.2.2.4. RowsRemoved

Удаление строки.

Параметры

first	int4	индекс первой удаленной строки
-------	------	--------------------------------

last	int4	индекс последней удаленной строки
------	------	-----------------------------------

12.2.3. Функции

12.2.3.1. GetCellData

Возвращает значение указанной ячейки.

Синтаксис

```
variant GetCellData(Row, Column)
```

Параметры

Row	int	порядковый номер строки. Нумерация начинается с 0
Column	int	порядковый номер столбца. Нумерация начинается с 0

12.2.3.2. GetRowDataAsJson

Возвращает JSON-объект с данными указанной строки.

Синтаксис

```
GetRowDataAsJson(Row)
```

Параметры

Row	int	порядковый номер строки. Нумерация начинается с 0
-----	-----	---------------------------------------------------

12.2.3.3. GetColumnCount

Возвращает количество столбцов, существующих в таблице на момент вызова функции.

Синтаксис

```
string GetColumnCount()
```

12.2.3.4. GetRowCount

Возвращает количество строк, существующих в таблице на момент вызова функции.

Синтаксис

```
int4 GetRowCount()
```

12.2.3.5. SetCellData

Задаёт значение указанной ячейке.

Синтаксис

```
void SetCellData(Row, Column, Value)
```

Параметры

Row	int	порядковый номер строки. Нумерация начинается с 0
Column	int	порядковый номер столбца. Нумерация начинается с 0
Value	variant	значение, добавляемое в ячейку

12.2.3.6. SetDataAsJson

Задаёт данные в формате JSON.

Синтаксис

```
void SetDataAsJson(Data)
```

Параметры

Data	string	объект в формате JSON
------	--------	-----------------------

12.2.3.7. SetRowDataAsJson

Задаёт данные в формате JSON.

Синтаксис

```
void SetRowDataAsJson(Row, Data)
```

Параметры

Row	int	порядковый номер строки. Нумерация начинается с 0
Data	string	данные, добавляемые в строку

12.2.3.8. AddColumn

Добавляет новый столбец в таблицу модели данных.

Синтаксис

```
void AddColumn(ColumnName, Title, Description, EUnit, Type, ReadOnly, Emptiable)
```

Параметры

ColumnName	string	<p>идентификатор столбца, возможные значения:</p> <ul style="list-style-type: none"> ➤ «source» - источник события ➤ «time» - время генерации уведомления о событии ➤ «message» - сообщение <ul style="list-style-type: none"> ➤ «source» - источник события ➤ «time» - время генерации уведомления о событии ➤ «message» - сообщение ➤ «severity» - уровень важности события ➤ «condition_name» - имя условия генерации события ➤ «subcondition_name» - имя подусловия генерации события ➤ «quality» - текущее качество сигнала, изменение которого привело к генерации события ➤ «active_time» - время перехода состояния события в активное ➤ «actor_id» - имя пользователя, выполнившего квитирование сообщения о событии ➤ «ack» - признак квитирования события ➤ «active» - признак активности подусловия, по которому было сгенерировано событие ➤ «cookie» - специальный идентификатор события, который имеет служебное назначение. Необходим для сведения квитанций, деактиваций с событиями ➤ «ack_time» - время квитирования ➤ «ack_comment» - комментарий квитирования ➤ «ack_required» - требование квитирования ➤ «value» - значение сигнала, изменение которого привело к генерации события ➤ «sound» - звуковой файл, исполняемый при выполнении подусловия генерации сообщения о событии ➤ «area_path» - относительный тег объекта ➤ «object_id» - идентификатор объекта на сервере, по которому было сгенерировано событие ➤ «deactive_time» - время, когда событие перешло из активного состояния в неактивное
Title	string	заголовок столбца
Description	string	описание столбца
EUnit	string	единицы измерения значения в столбце
Type	string	тип данных значения в столбце
ReadOnly	bool	модификатор доступа "только чтение"
Emptiable	bool	разрешение/запрет ячейкам в столбце не содержать значений

12.2.3.9. AddRow

Добавляет новую строку в конец таблицы модели данных.

Синтаксис

```
void AddRow()
```

12.2.3.10. InsertRow

Добавляет новую строку после указанной строки в таблице модели данных.

Синтаксис

```
void InsertRow(Row)
```

Параметры

Row	int	порядковый номер строки, после которой будет вставлена новая строка. Нумерация начинается с 0
-----	-----	-----------------------------------------------------------------------------------------------

12.2.3.11. InsertRows

Добавляет несколько новых строк после указанной строки в таблице модели данных.

Синтаксис

```
void InsertRows(Row, Count)
```

Параметры

Row	int	порядковый номер строки, после которой будут вставлены новые строки. Нумерация начинается с 0
Count	int	количество добавляемых строк

12.2.3.12. RemoveRow

Удаляет указанную строку из таблицы модели данных.

Синтаксис

```
void RemoveRow(Row)
```

Параметры

Row	int	порядковый номер строки. Нумерация начинается с 0
-----	-----	---------------------------------------------------

12.2.3.13. RemoveRows

Удаляет указанный диапазон строк.

Синтаксис

```
void RemoveRows(Index1, Index2)
```

Параметры

Index1	int	индекс первой строки из списка удаляемых
Index2	int	индекс последней строки из списка удаляемых

12.2.3.14. BeginReadAsync

Заполняет таблицу модели данными из источника.

Синтаксис

```
void BeginReadAsync()
```

12.2.3.15. CancelReadAsync

Прерывает чтение данных из источника.

Синтаксис

```
void CancelReadAsync()
```

12.2.3.16. Clear

Удаляет все строки из таблицы модели данных.

Синтаксис

```
void Clear()
```

12.2.3.17. Reset

Удаляет все строки и столбцы из таблицы модели данных.

Синтаксис

```
void Reset()
```

12.3. Таблица: столбец

Компонент позволяет сформировать таблицу для отображения данных из модели.

12.3.1. Свойства

12.3.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

12.3.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

12.3.1.3. Длина

```
int8 Length
```

Размер массива (количество элементов в массиве).

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

12.3.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```


**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){
  TextEdit_1.Text = "Включен режим 1";}
else
{TextEdit_1.Text = "Включен режим 2";}
```

12.3.1.5. Заголовок

```
string Title
```

Заголовок столбца.

12.3.1.6. Описание

```
string Description
```

Описание столбца.

12.3.1.7. Идентификатор

```
string Id
```

Идентификатор столбца.

Значение

«source»	источник события
«time»	время генерации уведомления о событии
«message»	сообщение
«severity»	уровень важности события

«condition_name»	имя условия генерации события
«subcondition_name»	имя подусловия генерации события
«quality»	текущее качество сигнала, изменение которого привело к генерации события
«active_time»	время перехода состояния события в активное
«actor_id»	имя пользователя, выполнившего квитирование сообщения о событии
«ack»	признак квитирования события
«active»	признак активности подусловия, по которому было сгенерировано событие

«cookie»	специальный идентификатор события, который имеет служебное назначение. Необходим для сведения квитанций, деактиваций с событиями
«ack_time»	время квитирования
«ack_comment»	комментарий квитирования
«ack_required»	требование квитирования
«value»	значение сигнала, изменение которого привело к генерации события
«sound»	звуковой файл, исполняемый при выполнении подусловия генерации сообщения о событии
«area_path»	относительный тег объекта
«object_id»	идентификатор объекта на сервере, по которому было сгенерировано событие

«deactive_
time»

время, когда
событие
перешло из
активного
состояния в
неактивное

12.3.1.8. Цвет фона

uint4 BackgroundColor

Цвета фона столбца.

12.3.1.9. Цвет текста

uint4 ForegroundColor

Цвет текста в столбце.

12.3.1.10. Шрифт текста

string Font

Шрифт текста в столбце.

12.3.1.11. Выравнивание текста

int4 TextAlignment

Вид выравнивания текста в столбце.

12.3.1.12. Выравнивание текста

int4 TextAlignment

Вид выравнивания текста в столбце.

12.3.1.13. Цвет фона заголовка

uint4 TitleBackgroundColor

Цвет фона заголовка столбца.

12.3.1.14. Цвет текста заголовка

```
uint4 TitleForegroundColor
```

Цвет текста заголовка столбца.

12.3.1.15. Шрифт текста заголовка

```
uint4 TitleFont
```

Шрифт текста заголовка столбца.

12.3.1.16. Ширина

```
int4 Width
```

Ширина столбца (в пикселях).

12.3.1.17. Режим изменения ширины

```
uint4 WidthResizeMode
```

Изменение ширины столбца при изменении содержимого.

Значение

«0»	ручное изменение ширины с помощью мыши
«1»	максимальное растяжение столбца по полю таблицы
«2»	фиксированная ширина столбца, заданная в свойствах столбца
«3»	автоматическое изменение ширины столбца в соответствии с шириной содержимого

12.3.1.18. Может не иметь значения

```
bool Emptiable
```

Разрешает/запрещает ячейкам столбца не содержать значений.

Значение

«true»	ячейки могут быть пустыми
«false»	в ячейки следует записать значения

12.3.1.19. Скрывать

```
bool Hidden
```

Скрывает/отображает столбец.

Значение

«true»	столбец скрыт
«false»	столбец отображается

12.3.1.20. Числовой формат

```
bool NumberFormat
```

Формат поступающих числовых данных.

12.3.1.21. Временной формат

```
string TimeFormat
```

Формат выводимой даты полученных данных.

12.3.1.22. Фильтр в JSON-формате

```
string JsonFilter
```

Условия фильтрации отображаемых данных в таблице. Задаются в виде строки - объекта JSON.

Правила формирования условий фильтрации

Фильтрация возможна только для столбцов с идентификаторами (переменными): «**severity**», «**message**», «**source**», «**area**», «**acked**».

Условие фильтрации представляет собой одиночный ({"ключ": "значение"}) или составной (состоящий из нескольких объектов) объект JSON.

Составной объект формируется с помощью:

- логических операций:
 - конъюнкция - "and" или "&")
 - дизъюнкция - "or" или "|")
 - инверсия - "not" или "!"
- операций сравнения:
 - больше - ">"
 - больше либо равно - ">="
 - меньше - "<"
 - меньше либо равно - "<="
 - соответствует - "=="
 - не соответствует - "!="

В зависимости от используемой операции составной объект JSON может иметь формат:

- {"and": [ARRAY]}
- {"or": [ARRAY]}
- {"not": { OBJECT }}
- {"операция сравнения": { VARIABLE }} (операцию сравнения "==" можно не указывать, т.е. формат объекта - { VARIABLE })

где:

- VARIABLE - переменная. Обозначается структурой <имя переменной>: <значение переменной>
- OBJECT - переменная или составной объект JSON любого из указанных выше форматов
- ARRAY - массив. Может состоять из переменных и составных объектов JSON (любого формата) в различных комбинациях

Примеры



ПРИМЕР

Отфильтровать по содержимому столбца с числовыми данными (например, важность). Отобразить события с важностью 700.

При фильтрации по столбцу имя переменной должно быть пустым, т.е. указанное условие имеет вид:

```
{"": {"": "700"}}
```



ПРИМЕР

Отфильтровать по содержимому столбца с текстовыми данными. Отобразить события, содержащие текст "поставлен" или "снят".

```
{
  "or": [
    {"": "*поставлен*"},
    {"": "*снят*"}
  ]
}
```



ПРИМЕР

Отфильтровать данные по содержимому столбца с числовыми данными (например, важность).
Отобразить события с важностью от 100 до 300.

```
{
  "and": [
    {">=": {"": 100}},
    {"<=": {"": 300}}
  ]
}
```

12.3.1.23. Сортировать

```
bool SortingEnabled
```

Разрешение/запрет сортировки данных в столбце.

12.3.1.24. Порядок сортировки

```
uint1 SortOrder
```

Вид сортировки данных в столбце.

Значение

«0»	по возрастанию (прямо по алфавиту)
«1»	по убыванию (обратно по алфавиту)

12.3.1.25. Приоритет сортировки

```
uint4 SortPriority
```

Порядок применения сортировки.

12.3.1.26. Разрешить перенос слов

```
bool WordWrap
```

Разрешение/запрет переноса слов в столбце.

Значение

«true»	перенос слов разрешен
«false»	перенос слов запрещен

12.3.2. События

12.3.2.1. MouseClick

Щелчок левой кнопкой мыши по ячейке.

Параметры

row	int4	порядковый номер строки, по ячейке которой был выполнен щелчок
-----	------	----------------------------------------------------------------

12.3.2.2. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по ячейке.

Параметры

row	int4	порядковый номер строки, по ячейке которой был выполнен двойной щелчок
-----	------	------------------------------------------------------------------------

12.3.2.3. MouseRightClick

Щелчок правой кнопкой мыши по ячейке столбца.

Параметры

row	int4	порядковый номер строки, по ячейке которой был выполнен щелчок
-----	------	----------------------------------------------------------------

12.3.2.4. MouseTitleClick

Щелчок левой кнопкой мыши по заголовку столбца.

12.3.2.5. MouseRightTitleClick

Щелчок правой кнопкой мыши по заголовку столбца.

12.3.2.6. WidthChanged

Изменение ширины столбца.

12.3.3. Функции

12.3.3.1. GetIndex

Возвращает индекс столбца в таблице.

Синтаксис

```
int4 GetIndex()
```

12.3.3.2. GetCellData

Возвращает значение указанной ячейки.

Синтаксис

```
string GetCellData(Cell)
```

Параметры

Cell	int	порядковый номер ячейки
------	-----	-------------------------

12.4. Таблица: декоратор

Компонент позволяет выделять цветом отдельные строки или столбцы таблицы.

12.4.1. Свойства

12.4.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

12.4.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
-----	------------------

>«1»	массив соответствующей размерности
------	------------------------------------

12.4.1.3. Настройки декоратора

```
string JsonDecorator
```

Задаются в виде строки - объекта JSON.

12.5. Таблица: итератор

Компонент позволяет выполнять перебор строк от первой строки до последней. Устаревший компонент, вместо него используйте компонент Таблица: устойчивый итератор.

12.5.1. Свойства

12.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

12.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

12.5.1.3. Индекс текущей строки

```
int4 CurrentRow
```

Порядковый номер строки, с которой будет проведена обработка данных.

12.5.2. Функции

12.5.2.1. FirstCelected

Инициализирует итератор.

Синтаксис

```
FirstCelected()
```

12.5.2.2. FirstCustom

Инициализирует итератор с пользовательским фильтром.

Синтаксис

```
FirstCustom(Filter)
```

Параметры

Filter	string	наименование фильтра
--------	--------	----------------------

12.5.2.3. Next

Переходит к следующей строке.

Синтаксис

```
Next()
```

12.5.2.4. End

Возвращает true, если достигнут конец итерации.

Синтаксис

```
End()
```

12.6. Таблица: устойчивый итератор

Компонент расширяет возможности компонента **Таблица: итератор**. Позволяет выполнять перебор строк от первой строки до последней, а также перебор выделенных строк таблицы.

12.6.1. Свойства

12.6.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

12.6.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

12.6.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

12.6.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

**ПРИМЕЧАНИЕ**

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры

**ПРИМЕР**

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```

**ПРИМЕР**

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```

**ПРИМЕР**

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
{TextEdit_1.Text = "Включен режим 2";}
```

12.6.2. Функции

12.6.2.1. GetData

Возвращает значение (с типом variant) ячейки строки, на которую указывает итератор.

Синтаксис

```
variant GetData(Index)
```

Параметры

Index	int	индекс строки, на которой установлен итератор
-------	-----	-----------------------------------------------

12.6.2.2. GetText

Возвращает значение ячейки строки, на которую указывает итератор, в виде текстовой строки.

Синтаксис

```
string GetText(Index)
```

Параметры

Index	int	индекс строки, на которой установлен итератор
-------	-----	-----------------------------------------------

12.6.2.3. GetRow

Возвращает номер текущей строки.

Синтаксис

```
int4 GetRow()
```

12.6.2.4. GetStatus

Возвращает код состояния строки, на которую указывает итератор.

Синтаксис

```
uint1 GetStatus()
```

12.6.2.5. IsRowSelected

Возвращает true, если текущая строка выделена.

Синтаксис

```
bool IsRowSelected()
```

12.6.2.6. SetRowSelected

Выделяет или снимает выделение с текущей строки.

Синтаксис

```
void SetRowSelected(FlagBool)
```

Параметры

FlagBool	bool	флаг выделения/снятия выделения
----------	------	---------------------------------

12.6.2.7. FirstRow

Инициализирует итератор для прохода по всем строкам таблицы, начиная с первой.

```
void FirstRow()
```

12.6.2.8. FirstSelectedRow

Инициализирует итератор для прохода по выделенным строкам таблицы, начиная с первой выделенной.

```
void FirstSelectedRow()
```

12.6.2.9. Next

Переходит к следующей строке, в зависимости от режима (проход по выделенным или проход по всем).

```
void Next()
```

12.6.2.10. HasNext

Выводит true, если текущая строка является последней для выбранного режима итератора.

```
bool HasNext()
```

12.6.2.11. RemoveRow

Удаляет текущую строку из таблицы.

```
void RemoveRow()
```

12.6.2.12. Close

Деинициализирует итератор.


```
void Close()
```

13. Экранные формы

Состав экранной формы:

- **Контейнер экранной формы** – базовый элемент, создаваемый вместе с экранной формой. Является объектом-контейнером для любых элементов экранной формы.
- **Окно экранной формы** – родительский элемент для контейнера экранной формы.

Чтобы обратиться к свойствам экранной формы из кода, используйте контекст [here](#).



ПРИМЕЧАНИЕ

Если размеры контейнера превышают размеры окна, то в режиме исполнения появятся полосы прокрутки. Если размеры окна превышают размеры контейнера, то в режиме исполнения будут видны пустые области.

Чтобы автоматически подогнать размеры контейнера экранной формы к размерам окна экранной формы, используйте свойство [Режим масштабирования](#).

13.1. Свойства

13.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

13.1.2. X и Y



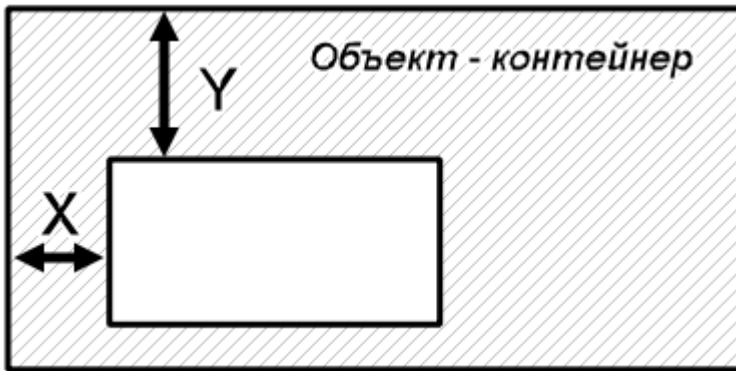
ОБРАТИТЕ ВНИМАНИЕ

Свойства **X** и **Y** заблокированы в редакторе свойств. Чтобы изменить значения данных свойств, воспользуйтесь скриптом обработчика событий.

Float X

Float Y

Позиция объекта по горизонтали и по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).



Примеры



ПРИМЕР

Установить позицию элемента (100; 100).

```
Object.X = 100;  
Object.Y = 100;
```



ПРИМЕР

Сдвинуть объект на 50 пикселей правее и ниже его текущей позиции.

```
Object.X += 50;  
Object.Y += 50;
```

13.1.3. Угол поворота



ОБРАТИТЕ ВНИМАНИЕ

Свойство **Угол поворота** заблокировано в редакторе свойств. Чтобы изменить значения данного свойства, воспользуйтесь скриптом обработчика событий.

Float Rotation

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.

Примеры



ПРИМЕР

Задать угол поворота объекта - 45 градусов.

```
Object.Rotation = 45;
```



ПРИМЕР

Увеличить угол поворот объекта на 1 градус относительно текущего значения

```
Object.Rotation += 1;
```

13.1.4. Масштаб

Float Scale

Коэффициент масштабирования контейнера экранной формы.

13.1.5. Видимость

bool Visible

Видимость элементов экранной формы.

Примеры



ПРИМЕР

Включить отображение

```
if(signal1.value > signal2.value){  
    here.Visible = true;  
}
```



ПРИМЕР

Отключить отображение

```
if(signal1.value < signal2.value){  
    here.Visible = false;  
}
```

13.1.6. Включено

bool Enabled

Активность объекта.

Button

Button

Значение

true	объект активен
false	объект заблокирован

Примеры



ПРИМЕР

Активировать объект в зависимости от выполнения условия.

```
if (Signal1.value == Signal2.value){
    Object.Enabled = true;
}
else{
    Object.Enabled = false;
}
```

13.1.7. Всплывающая подсказка

```
string Tooltip
```

Подсказка, всплывающая при наведении курсора на свободную область контейнера экранной формы.

Примеры



ПРИМЕР

Задать всплывающую подсказку для контейнера экранной формы:

```
here.Tooltip = "Подача команды MNA_start = true";
```

13.1.8. Ширина и высота

```
float Width
```

```
float Height
```

Ширина и высота контейнера экранной формы в пикселях.

Примеры



ПРИМЕР

Задать ширину контейнера экранной формы

```
here.Width = 50;
```



ПРИМЕР

Уменьшить текущую ширину контейнера экранной формы на 50 пикселей

```
here.Width -= 50;
```



ПРИМЕР

Задать высоту контейнера экранной формы

```
here.Height = 500;
```




ПРИМЕР

Увеличить текущую высоту контейнера экранной формы вдвое

```
here.Height *= 2;
```

13.1.9. Цвет пера

uint4 PenColor

Цвет внешней границы контейнера экранной формы. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления .

Примеры



ПРИМЕР

Задать черный цвет пера






```
here.PenColor = 0xff000000; //в шестнадцатеричной системе счисления  
here.PenColor = 4278190080; //в десятичной системе счисления
```

13.1.10. Стиль пера

uint2 PenStyle

Стиль внешней границы контейнера экранной формы.

Значение

Значение	Граница	Внешний вид
«0»	Без границы	
«1»	Сплошная линия	
«2»	Длинный пунктир	
«3»	Короткий пунктир	
«4»	Смешанный пунктир	
«5»	Смешанный пунктир	

Примеры



ПРИМЕР

Применить стиль границы (короткий пунктир) для объекта

```
here.PenStyle = 3;
```

13.1.11. Толщина пера

```
float PenWidth
```

Толщина внешней границы контейнера экранной формы. Измеряется в пикселях.

Примеры

ПРИМЕР

Задать толщину пера 5 пикселей

```
here.PenWidth = 5;
```

13.1.12. Цвет заливки

```
uint4 BrushColor
```

Цвет заливки окна экранной формы. Указывается через системное окно **Выбор цвета**. После выбора автоматически переводится в шестнадцатеричную систему счисления

Примеры

ПРИМЕР

Залить объект красным цветом

```
here.BrushColor = 0xffff0000; //в шестнадцатеричной системе счисления
here.BrushColor = 4294901760; //в десятичной системе счисления
```

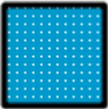

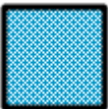

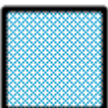
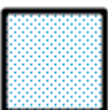
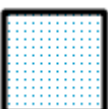

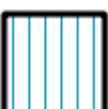
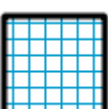

13.1.13. Стил ь заливки



```
uint2 BrushStyle
```

Заливает объект одним из стилей заливки.

Значение

Значение	Стил ь	Внешний вид
«0»	Нет заливки	
«1»	Сплошная заливка	

Значение	Стиль	Внешний вид
«2»	Точечная 1	
«3»	Точечная 2	
«4»	Точечная 2	
«5»	Точечная 4	
«6»	Точечная 5	
«7»	Точечная 6	
«8»	Точечная 7	
«9»	Горизонтальные линии	
«10»	Вертикальные линии	
«11»	Клетка	
«12»	Обратные диагональные линии	

Значение	Стиль	Внешний вид
«13»	Прямые диагональные линии	
«14»	Диагональная клетка	

Примеры



ПРИМЕР

Залить объект узором в клетку

```
Object.BrushStyle = 11;
```

13.1.14. X окна и Y окна

```
int4 WindowX
```

```
int4 WindowY
```

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству **X** для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.

Примеры



ПРИМЕР

Установить позицию окна экранной формы (10; 50)

```
here.WindowX = 10;  
here.WindowY = 50;
```

13.1.15. Ширина окна и Высота окна

```
int4 WindowWidth
```

```
int4 WindowHeight
```

Ширина и высота окна экранной формы в пикселях. Может отличаться от свойств **Ширина** или **Высота**, которые относятся к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.

Примеры



ПРИМЕР

Установить ширину и высоту окна (800x600 px)

```
here.WindowWidth = 800;  
here.WindowHeight = 600;
```

13.1.16. Заголовок окна

```
string WindowCaption
```

Надпись в строке заголовка.

Примеры



ПРИМЕР

Установить заголовок окна

```
here.WindowCaption = "Заголовок окна";
```

13.1.17. Показывать заголовок

```
bool ShowWindowCaption
```

Включает/отключает заголовок окна вместе со всеми кнопками.



Примеры



ПРИМЕР

Убрать заголовок окна

```
here.ShowWindowCaption = false;
```

13.1.18. Показывать кнопку "свернуть"

```
bool ShowWindowMinimize
```

Включает/отключает кнопку  в заголовке окна.

Примеры



ПРИМЕР

Убрать кнопку "свернуть"

```
here.ShowWindowMinimize = false;
```

13.1.19. Показывать кнопку "развернуть"

```
bool ShowWindowMaximize
```

Включает/отключает кнопку  в заголовке окна.

Примеры



ПРИМЕР

Убрать кнопку "развернуть"

```
here.ShowWindowMaximize = false;
```

13.1.20. Показывать кнопку "заккрыть"

```
bool ShowWindowClose
```

Включает/отключает кнопку  в заголовке окна.

Примеры



ПРИМЕР

Убрать кнопку "заккрыть"

```
here.ShowWindowClose = false;
```

13.1.21. Поверх всех окон

```
bool WindowSizeMode
```

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.

Примеры



ПРИМЕР

Установить отображение экранной формы поверх остальных окон

```
here.WindowSizeMode = true;
```

13.1.22. Размеры окна

```
uint4 WindowSizeMode
```

Определяет режим вычисления размеров окна.

Значение

«0»	вручную
«1»	автоподбор при открытии
«2»	автоподбор

Примеры



ПРИМЕР

Отобразить режим вычисления размеров окна в текстовом поле

```
TextEdit.Text = here.WindowSizeMode;
```

13.1.23. Стиль рамки окна

```
uint4 WindowBorderStyle
```

Стиль рамки окна экранной формы.

Значение

«0»	без рамки
«1»	изменяемый размер
«2»	фиксированный размер

Примеры



ПРИМЕР

Отобразить код текущего стиля рамки окна в текстовом поле

```
TextEdit.Text = here.WindowBorderStyle;
```

13.1.24. Состояние окна

```
uint4 WindowSizeMode
```

Определяет режим, в котором будет открываться окно экранной формы.

Значение

«0»	по умолчанию
«1»	развернуто
«2»	на весь экран
«3»	свернуто

Примеры



ПРИМЕР

Показать код текущего режима в текстовом поле

```
TextEdit.Text = here.WindowSizeMode;
```

13.1.25. Режим масштабирования

```
uint4 WindowScalingMode
```

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.

Значение

« 0 »	не масштабировать
« 1 »	всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
« 2 »	только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

Примеры



ПРИМЕР

Выключить режим масштабирования

```
here.WindowScalingMode = 0;
```



ПРИМЕР

Включить режим "Всегда масштабировать"

```
here.WindowScalingMode = 1;
```



ПРИМЕР

Включить режим "Только уменьшать"

```
here.WindowScalingMode = 2;
```

13.1.26. Монитор

```
int4 MonitorNumber
```

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



ОБРАТИТЕ ВНИМАНИЕ

Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства **Монитор** нужно оставлять равным 0.

Примеры



ПРИМЕР

Отобразить номер монитора в текстовом поле

```
TextEdit.Text = here.MonitorNumber;
```

13.1.27. Положение окна

uint4 WindowPosition

Определяет позицию для открытия окна экранной формы.

Значение

«0»	вручную
«1»	по центру монитора
«2»	по центру родительского окна
«3»	по центру родительской формы
«4»	вручную относительно монитора

Примеры



ПРИМЕР

Отобразить код позиции в текстовом поле

```
TextEdit.Text = here.WindowPosition;
```

13.1.28. Режим обработки закрытия окна

uint2 WindowCloseMode

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение «Отправить запрос». Свойство связано с функцией **Close()**.



ПРИМЕЧАНИЕ

Свойство игнорируется для окон, открытых с помощью элемента **Фрейм**.

Примеры



ПРИМЕР

Режим обработки закрытия окна - Отправить запрос

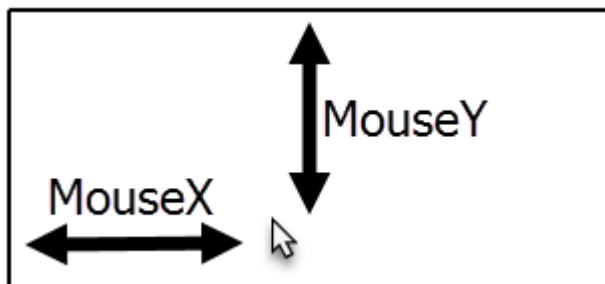
```
here.WindowCloseMode = 2;
```

13.1.29. MouseX и MouseY

```
float MouseX
```

```
float MouseY
```

Позиция курсора мыши в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Вывести в текстовую область координаты курсора мыши внутри объекта

```
TextEdit.Text = Str.ToString(Object.MouseX);  
TextEdit.Text = Str.ToString(Object.MouseY);
```

13.1.30. Активность окна

```
bool WindowActive
```

Текущее состояние активности окна экранной формы.



ПРИМЕЧАНИЕ

Доступно только для чтения в режиме рантайма.

Значение

«true»	ОКНО АКТИВНО
«false»	ОКНО НЕАКТИВНО

Примеры



ПРИМЕР

Вывести состояние активности окна в поле ввода

```
TextEdit.Text = Str.ToString(here.WindowActive);
```

13.2. События

13.2.1. MouseMove

Перемещение курсора внутри области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

13.2.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

13.2.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

13.2.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события

KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события
-------------------	------	-----------------------------------------------------------------------

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

13.2.5. MousePress

Нажатие кнопки мыши в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

13.2.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

13.2.7. MouseEnter

Ввод курсора в область объекта.

Параметры

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;  
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

13.2.8. MouseLeave

Вывод курсора за пределы области объекта.

Параметры

Для доступа к параметрам используйте структуру event.

X	Float	локальная координата объекта по оси X, в которой возникло событие
Y	Float	локальная координата объекта по оси Y, в которой возникло событие

Buttons	Int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	Int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

Примеры



ПРИМЕР

Разместить кнопку в месте срабатывания события.

```
Button.X = event.X;
Button.Y = event.Y;
```



ПРИМЕР

Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события.

```
TextEdit_1.Text = Str.ToString(event.KeyboardModifiers);
```



ПРИМЕР

Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события.

```
TextEdit_2.Text = Str.ToString(event.Buttons);
```

13.2.9. CloseRequested

Запрос на закрытие формы.

13.2.10. Opened

Открытие формы.

13.2.11. Closed

Закрытие формы.

13.2.12. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

Параметры

active	bool	статус активности окна: активное («true») или неактивное («false»)
--------	------	--------------------------------------------------------------------

13.2.13. WindowSizeChanged

Изменение размеров окна.

Параметры

width	uint4	новая ширина окна
height	uint4	новая высота окна

13.2.14. WindowPosChanged

Изменение положения окна.

Параметры

x	int4	новое положение по горизонтали верхней левой точки окна
y	int4	новое положение по вертикали верхней левой точки окна

13.2.15. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

Параметры

KeyCode	uint4	код клавиши
Modifiers	int4	код клавиши-модификатора
Repeated	bool	флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: «Ctrl», «Shift», «Alt»
Text	string	текст клавиши

Примеры



ПРИМЕР

Вывести код нажатой клавиши в текстовое поле

```
TextEdit_1.Text = Str.ToString(event.KeyCode);
```



ПРИМЕР

Вывести код нажатой клавиши-модификатора в текстовое поле

```
TextEdit_1.Text = Str.ToString(event.Modifiers);
```



ПРИМЕР

При удержании клавиши F5 записать текст из поля ввода в текстовое поле

```
if (event.KeyCode == 55 && event.Repeated == true)
{Text_1.Text = TextEdit_1.Text;}
```



ПРИМЕР

Вывести текст нажатой клавиши в текстовое поле

```
TextEdit_1.Text = event.Text;
```

13.2.16. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

Параметры

KeyCode	uint4	код клавиши
Modifiers	int4	код клавиши-модификатора
Repeated	bool	флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: «Ctrl», «Shift», «Alt»
Text	string	текст клавиши

Примеры



ПРИМЕР

Увеличить размер прямоугольника при отпуске пробела

```
if (event.KeyCode == 84)
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



ПРИМЕР

Уведомить о сохранении данных

```
if (event.Modifiers == 67108864 && event.KeyCode == 28)
{Text_1.Text = "Сохранено";}
```



ПРИМЕР

При удержании и отпуске клавиши F5 записать текст из поля ввода в текстовое поле

```
if (event.KeyCode == 55 && event.Repeated == true)
{Text_1.Text = TextEdit_1.Text;}
```



ПРИМЕР

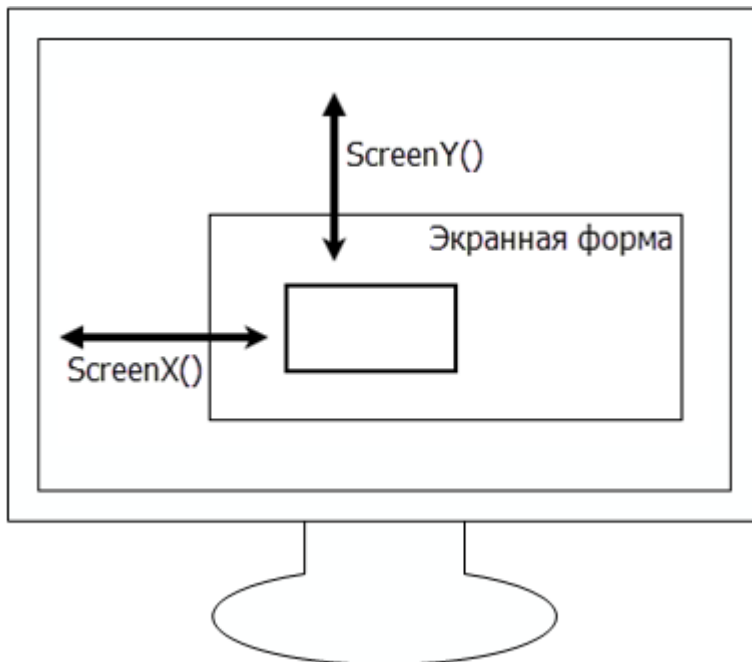
Вывести текст нажатой клавиши в текстовое поле

```
Text_1.Text = event.Text;
```

13.3. Функции

13.3.1. ScreenX, ScreenY

Вычисляют координаты объекта относительно экрана монитора. Координаты измеряются в пикселях.



Синтаксис

```
ScreenX(Object.X, Object.Y)
```

```
ScreenY(Object.X, Object.Y)
```

Параметры

Object.X	float	значение свойства X объекта
Object.Y	float	значение свойства Y объекта

Примеры



ПРИМЕР

Записать координаты X и Y объекта относительно экрана в поля ввода

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

13.3.2. Close

Закрывает экранную форму. Функция связана с свойством **Режим обработки закрытия окна**.

Синтаксис

```
close()
```

Примеры



ПРИМЕР

При выборе флажка закрыть форму, если флажок не выбран, вывести сообщение

```
if (Checkbox.State == true)  
    me.Close();  
else  
    TextEdit.Text = "Чтобы закрыть форму, установите флажок внизу";
```

13.3.3. SaveScreenShot

Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).

13.3.4. Синтаксис

```
SaveScreenShot(filePath)
```

Параметры

filePath	string	путь к файлу
----------	--------	--------------

Примеры



ПРИМЕР

Сделать скриншот формы

```
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

13.3.5. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.

Синтаксис

```
DoAutoSize()
```

Примеры



ПРИМЕР

Установить размер окна по размеру экранной формы

```
here.DoAutoSize();
```

13.3.6. SetBoundingRegion

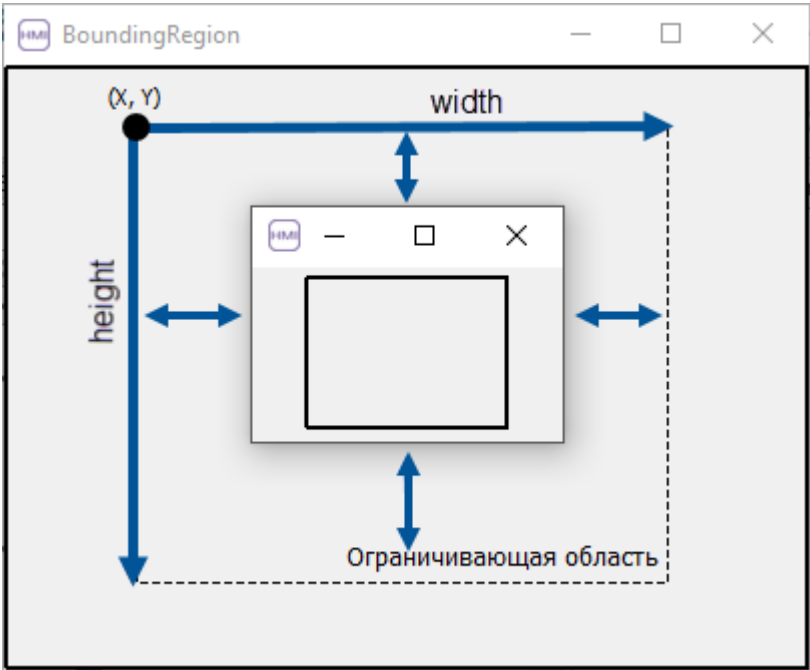
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.

Синтаксис

```
SetBoundingRegion(int x, int y, uint width, uint height)
```

Параметры

x, y	int	координаты начала области ограничения
width, height	uint	ширина и высота области ограничения, отсчитываются от точки начала области



Примеры

ПРИМЕР

Установить ограничения перемещения окна

```
here.SetBoundingRegion(500, 200, 400, 400);
```

13.3.7. ResetBoundingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.

Синтаксис

```
ResetBoundingRegion()
```

Примеры



ПРИМЕР

Отменить ограничения перемещения окна

```
ResetBoundingRegion();
```

14. Средство отладки

Компонент позволяет вести журнал времени исполнения кода на языке SePlatform.Om.

14.1. Свойства

14.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

14.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

14.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```


14.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){  
    TextEdit_1.Text = "Включен режим 1";}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

14.2. Функции

14.2.1. Log

Выводит сообщение в Журнал времени исполнения.

Синтаксис

```
Log(variant)
```

Параметры

variant	текст сообщения
---------	-----------------

Примеры



ПРИМЕР

Вывести сообщение о завершении отладки:

```
DebugTool.Log("Отладка завершена");
```

15. Менеджер проекта

Компонент позволяет без перезапуска обновлять проект SePlatform.HMI, работающий в режиме исполнения.

15.1. Свойства

15.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

15.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

15.2. Функции

15.2.1. AsyncReload

Заменяет файлы текущего запущенного проекта на файлы из указанной папки обновлений, компилирует и запускает обновленный проект. При перезагрузке проекта также сохраняет копию текущего проекта.

Синтаксис

```
void AsyncReload(pathUpdate, pathBackup)
```

Параметры

pathUpdate	путь к проекту, изменения которого следует загрузить в запущенный проект
pathBackup	путь к папке для автосохранения копии текущего проекта

Примеры



ПРИМЕР

Перезагрузить проект.

```
ProjectManager_1.AsyncReload("C:\\Backup\\Update\\_BackupManager.hmi",  
"C:\\Backup\\Back");
```

15.2.2. AsyncReloadFromBackup

Загружает проект из резервной копии, компилирует и запускает восстановленный проект.

Синтаксис

```
void AsyncReloadFromBackup(pathBackup)
```

Параметры

pathBackup	путь к папке, из которой следует загрузить резервную копию проекта
------------	--------------------------------------------------------------------

Примеры



ПРИМЕР

Откатить изменения проекта.

```
ProjectManager_1.AsyncReloadFromBackup("C:\\Backup\\Back");
```

15.2.3. CheckForUpdates

Проверяет наличие обновлений для проекта по указанному пути.

Синтаксис

```
bool CheckForUpdates(pathUpdate)
```

Параметры

pathUpdate	Путь к файлу проекта, для которого следует проверить обновления.
------------	------------------------------------------------------------------

Возвращаемое значение

«true»	В указанной папке есть проект с расширением .hmi или файл *.binom.
«false»	В указанной папке отсутствуют проект с расширением .hmi и файл *.binom.

Примеры



ПРИМЕР

Проверить наличие обновлений проекта в папке.

```
Text_2 = ProjectManager_1.CheckForUpdates("C:\\Backup\\Update\\_BackupManager.hmi") ?  
"true" : "false";
```

15.3. События

15.3.1. ReloadStarted

Запуск перезагрузки (обновления) проекта.

15.3.2. ReloadStarted

Завершение удачной попытки перезагрузки (обновления) проекта.

15.3.3. ReloadFailed

Завершение неудачной попытки перезагрузки (обновления) проекта.

Параметры

errorMessage	string	текст ошибки
--------------	--------	--------------



ПРИМЕР

Вывести текст ошибки при неудачной перезагрузке проекта.

```
Text_3 += "Failed Reload: " + errorMessage + "\r";
```

16. Среда исполнения

Компонент позволяет определить режим работы приложения SePlatform.HMI : «**desktop**» или «**web**».

16.1. Свойства

16.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

16.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

Значение

«1»	одиночный объект
>«1»	массив соответствующей размерности

16.1.3. Длина

int8 Length

Размер массива (количество элементов в массиве).



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле длину массива.

```
TextEdit_1.Text = Str.ToString (SW_1.Length);
```

16.1.4. Индекс

int8 Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



ПРИМЕЧАНИЕ

Есть только у массивов и доступно только для чтения в режиме рантайма.

Примеры



ПРИМЕР

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения.

```
TextEdit_2.Text = Str.ToString(Index);
```



ПРИМЕР

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения.

```
elseTextEdit_3.Text = Str.ToString(penwidth[Index]);
```



ПРИМЕР

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки.

```
if (SW_1.Mode.Index == 0){
    TextEdit_1.Text = "Включен режим 1";}
else
    {TextEdit_1.Text = "Включен режим 2";}
```

16.1.5. Режим работы

uint1 WorkMode

Текущий режим работы приложения SePlatform.HMI.

Значение

«0»	SePlatform.HMI работает в режиме «desktop» - приложение запущено локально на компьютере.
-----	------------------------------------------------------------------------------------------

«1»	SePlatform.HMI работает через веб-интерфейс.
-----	----------------------------------------------

Примеры



ПРИМЕР

Вывести информацию о режиме работы приложения в текстовое поле:

```
Text_1.Text = Runtime_1.WorkMode == 0 ? "desktop" : "web";
```